



UNIVERZITET U NOVOM SADU  
PRIRODNO – MATEMATIČKI FAKULTET  
DEPARTMAN ZA MATEMATIKU I INFORMATIKU



Boban Vesin

# **Personalizacija procesa elektronskog učenja u tutorskom sistemu primenom tehnologija semantičkog veba**

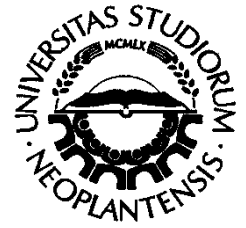
- doktorska disertacija -

Mentor:  
dr Mirjana Ivanović

Novi Sad, 2014.



UNIVERSITY OF NOVI SAD  
FACULTY OF SCIENCES  
DEPARTMENT OF  
MATHEMATICS AND INFORMATICS



Boban Vesin

# **Personalization of Learning Process in Tutoring System Supported with the Semantic Web Technologies**

- Doctoral Dissertation -

Mentor:  
PhD Mirjana Ivanović

Novi Sad, 2014.

## Sadržaj

1 UVOD .....	9
2 UČENJE UZ UPOTREBU VEBA .....	13
2.1 Elektronsko učenje .....	14
2.1.1 Organizacija materijala za elektronsko učenje .....	15
2.2 Popularnost elektronskog učenja .....	15
2.2.1 Inteligencija i adaptivnost sistema za elektronsko učenje.....	16
2.2.2 Inteligentni tutorski sistemi .....	16
3 PERSONALIZACIJA I ADAPTACIJA U SISTEMIMA UČENJA .....	18
3.1 Osnove personalizacije.....	19
3.1.1 Tipovi personalizacije .....	19
3.2 Identifikacija stilova učenja .....	21
3.2.1 Indeks stilova učenja .....	22
3.3 Sistemi za generisanje preporuka .....	23
3.3.1 Klasterovanje podataka i korisnika .....	24
3.3.2 Izbor optimalne putanje kretanja kroz nastavni materijal.....	24
3.4 Personalizacija prilagođavanjem linkova .....	25
3.4.1 Adaptivna hipermedija.....	25
3.5 Modeliranje individualnih karakteristika učenika.....	26
3.5.1 Izgled modela učenika u sistemima za elektronsko učenje .....	27
4 SEMANTIČKI WEB .....	29
4.1 Ontologije .....	30
4.1.1 Prikaz ontologija .....	31
4.1.2 Cilj upotrebe ontologija.....	33
4.1.3 Oblasti primene ontologija.....	34
4.2 Jezici semantičkog veba .....	35
4.2.1 XML - eXtensible Markup Language.....	36
4.2.2 RDFS - Resource Description Framework Schema .....	36
4.2.3 OWL - Ontology Web Language .....	37
4.2.4 SPARQL – Simple Protocol and RDF Query Language .....	37

4.3 Grafička okruženja za razvoj ontologija .....	38
4.4 Obrazovne ontologije .....	38
4.5 Pravila adaptacije .....	40
4.5.1 SWRL.....	40
4.5.2 Protégé .....	41
4.5.3 Jess .....	41
4.6 Arhitektura semantičkih sistema za elektronsko učenje .....	41
5 POREĐENJE SISTEMA PROTUS 2.0 I SLIČNIH TUTORSKIH SISTEMA .....	46
5.1 Sistemi sa adaptacijom na individualne stilove učenja učenika .....	46
5.2 Sistemi sa ugrađenim podsystemima za generisanje preporuka .....	47
5.3 Tutorski sistemi sa elementima tehnologija semantičkog veba .....	48
5.3.1 Upotreba pravila adaptacije u tutorskim sistemima .....	50
5.4 Primeri tutorskih sistema .....	50
5.4.1 iWeaver .....	50
5.4.2 Multitutor.....	53
5.4.3 Personal Reader .....	54
5.5 Popularnost tehnologija semantičkog veba.....	57
5.5.1 Alati i jezici za razvoj sistema baziranih na tehnologijama semantičkog veba ..	58
5.5.2 Oblasti upotrebe ontologija .....	59
6 IZGRADNJA MODELA TUTORSKOG SISTEMA .....	60
6.1 Arhitektura sistema.....	61
6.2 Ontologije sistema .....	62
6.2.1 Osnovne komponente ontologija .....	63
6.2.2 Ontologija znanja iz domena (eng. <i>Domain ontology</i> ).....	63
6.2.3 Ontologija zadataka (eng. <i>Task ontology</i> ).....	66
6.2.4 Ontologija modela učenika (eng. <i>Learner model ontology</i> ).....	67
6.2.5 Ontologija nastavne strategije (eng. <i>Teaching strategy ontology</i> ).....	70
6.2.6 Ontologija korisničkog interfejsa (eng. <i>Interface ontology</i> ) .....	71
6.3 Pravila adaptacije .....	71
6.3.1 Sintaksa pravila adaptacije.....	72
6.3.2 Pravila adaptacije na osnovu identifikacije stilova učenja.....	74
6.3.3 Pravila za formiranje modela učenika.....	80
6.3.4 Adaptacija na osnovu pravila za generisanje preporuka .....	82
6.4 Razvoj kurseva iz različitih domena .....	85
7 DIZAJN I IMPLEMENTACIJA SISTEMA PROTUS 2.0 .....	87
7.1 Prethodne verzije sistema Protus .....	87
7.1.1 Sistem Mag.....	88
7.1.2 Sistem Protus .....	90
7.2 Razvoj programerskog kursa u sistemu Protus 2.0 .....	93
7.2.1 Korisnički interfejs za učenika .....	94
7.2.2 Korisnički interfejs za predavača.....	95
7.3 Razvoj ontologija za programerski kurs iz Java programskog jezika.....	97
7.3.1 Ontologija znanja iz domena.....	98
7.3.2 Ontologija modela učenika .....	100
7.3.3 Ontologija nastavne strategije .....	100
7.3.4 Ontologija zadataka i ontologija korisničkog interfejsa .....	102

7.4 Personalizacija učenja u sistemu.....	102
7.4.1 Adaptacija u sistemu Protus 2.0 na osnovu stilova učenja.....	103
7.4.2 Upotreba sistema za generisanje preporuka.....	104
7.4.3 Šabloni navigacije.....	105
7.4.4 Proces evaluacije znanja učenika.....	107
7.5 Upotreba i funkcionisanje sistema.....	107
7.5.1 Bodovanje upitnika i određivanje početnog stila učenja.....	110
7.5.2 Realizacija kursa za učenje osnova Java programskog jezika.....	113
7.6 Nastavni materijal u sistemu Protus 2.0.....	115
8 DOPRINOSI DISERTACIJE I ZAVRŠNA RAZMATRANJA.....	125
8.1 Metrike ontologija.....	128
9 ZAKLJUČAK.....	131

## Lista Tabela

Tabela 1. Karakteristike identifikacije stilova učenja na osnovu modela Felder-a i Soloman-a.....	22
Tabela 2. Uporedni prikaz karakteristika sistema.....	57
Tabela 3. Atributi modela učenika u klasama <i>Concept</i> i <i>Resource</i> .....	64
Tabela 4. Atributi klasa <i>Ontologije modela učenika</i> .....	69
Tabela 5. Identifikacija klastera na osnovu stilova učenja.....	83
Tabela 6. Funkcionalnosti tri verzije sistema .....	88
Tabela 7. Primer instance klase <i>Concept</i> .....	98
Tabela 8. Primer instance klase <i>Resource</i> .....	99
Tabela 9. Primer instance klase <i>Interaction</i> .....	100
Tabela 10. Primer instance klase <i>Condition</i> .....	101
Tabela 11. Primer instance klase <i>BehaviourPattern</i> .....	101
Tabela 12. Primer popunjene tabele za određivanje stilova učenja.....	110
Tabela 13. Uporedni prikaz ontologija.....	129

## Lista ilustracija

Slika 1. Tradicionalna arhitektura inteligentnih tutorskih sistema .....	17
Slika 2. Grafička reprezentacija modela učenika.....	27
Slika 3. Ontologija sistema za elektronsko učenje prikazana kao semantička mreža.....	32
Slika 4. UML dijagram klasa sistema za elektronsko učenje .....	32
Slika 5. Ontologija sistema za elektronsko učenje predstavljena u jeziku OWL .....	33
Slika 6. Slojevi semantičkog veba .....	36
Slika 7. Editor za razvoj ontologija Protégé.....	38
Slika 8. Okruženje za učenje u semantičkom vebu .....	42
Slika 9. Koncepti sistema za učenje uz korišćenje semantičkog veba.....	43
Slika 10. Model obrazovnog servera.....	44
Slika 11. Korisnički interfejs sistema iWeaver .....	51
Slika 12. Arhitektura sistema <i>iWeaver</i> .....	52
Slika 13. Primer ontologije modela učenika Multitutora .....	53
Slika 14. Personal Reader .....	54
Slika 15. Ontologija dokumenata sistema <i>Personal Reader</i> .....	55
Slika 16. Ontologija profila učenika sistema <i>Personal Reader</i> .....	55
Slika 17. Ontologija za praćenje korisničkih sesija.....	56
Slika 18. Ontologija strukture elemenata sistema.....	56
Slika 19. Upotreba editora ontologija.....	58
Slika 20. Ontološki jezici trenutno u upotrebi .....	58
Slika 21. Industrija ili domen za koji ispitanici razvijaju ontologije.....	59
Slika 22. Arhitektura opšteg modela tutorskog sistema.....	61

Slika 23. Ontologije i pravila adaptacije u modelu tutorskog sistema .....	62
Slika 24. Nastavni materijal u sistemu Protus 2.0 .....	64
Slika 25. Topologija ontologije resursa opšteg modela tutorskog sistema .....	65
Slika 26. Ontologija zadataka sistema opšteg modela tutorskog sistema.....	66
Slika 27. Nivoi u modelu učenika .....	68
Slika 28. Ontologija modela učenika opšteg modela tutorskog sistema.....	68
Slika 29. Ontologija praćenja i modelovanja učenika u opštem modelu.....	70
Slika 30. Ontologija nastavne strategije sistema opšteg modela tutorskog sistema .....	70
Slika 31. Ontologija korisničkog interfejsa opšteg modela tutorskog sistema .....	71
Slika 32. Korisnički intefejs za učenike sa aktivnim stilom učenja.....	76
Slika 33. Korisnički intefejs za učenike sa reflektivnim stilom učenja .....	77
Slika 34. Preporuka opcije <i>Communication</i> .....	77
Slika 35. Preporuka opcije <i>Additional material</i> .....	77
Slika 36. Preporuka opcije <i>Syntax rules</i> .....	77
Slika 37. Primer lekcije za učenike sa vizuelnim stilom učenja.....	79
Slika 38. Primer lekcije za učenike sa verbalnim stilom učenja .....	79
Slika 39. Navigacija za učenike sa sekvencijalnim stilom učenja .....	80
Slika 40. Elementi za učenike sa globalnim stilom učenja.....	80
Slika 41. Opcije za promenu prikaza lekcije.....	80
Slika 42. Razvoj kurseva iz različitih domena u opštem modelu tutorskog sistema .....	85
Slika 43. Aktivnosti učenika u opštem modelu tutorskog sistema .....	86
Slika 44. Arhitektura sistema Mag.....	88
Slika 45. Administrativni deo sistema Mag.....	89
Slika 46. Korisnički interfejs sistema Mag.....	90
Slika 47. Arhitektura sistema Protus.....	91
Slika 48. Primer tutorijala .....	92
Slika 49. Prikaz primera u Protusu .....	92
Slika 50. Interfejs za predavača sistema Protus.....	93
Slika 51. Korisnički interfejs sistema Protus 2.0 .....	94
Slika 52. Mapa sajta sistema Protus 2.0 .....	95
Slika 53. Uvodni prozor aplikacije .....	95
Slika 54. Osnovni prozor administrativnog dela sistema.....	96
Slika 55. Pregled postojećeg nastavnog materijala .....	96
Slika 56. Unos novog nastavnog materijala.....	97
Slika 57. Deo ontologije znanja iz domena u sistemu Protus 2.0 .....	98
Slika 58. Jpg resurs sistema Protus 2.0 .....	99



Slika 59. Adaptacija na osnovu stilova učenja .....	103
Slika 60. Interakcija učenika i sistema Protus 2.0 .....	104
Slika 61. Adaptacija na osnovu navigacije kroz nastavni materijal.....	106
Slika 62. Adaptacija korisničkog interfejsa .....	106
Slika 63. Korisnička forma za prijavljivanje učenika .....	108
Slika 64. Registraciona forma sistema Protus 2.0.....	109
Slika 65. ILS upitnik .....	109
Slika 66. Izveštaj o rezultatima upitnika .....	112
Slika 67. Početna stranica sistema Protus 2.0 .....	113
Slika 68. Hijerarhija lekcija u kursu iz Java programskog jezika.....	113
Slika 69. Proces učenja u sistemu Protus 2.0.....	114
Slika 70. Korisnički interfejs za učenika .....	115
Slika 71. Hijerarhija nastavnog materijala .....	115
Slika 72. Grupisanost lekcija po jedinicama.....	116
Slika 73. Resurs za prikaz uvoda lekcije <i>If naredba</i> .....	116
Slika 74. Resurs za prikaz osnovnih informacija lekcije <i>If naredba</i> .....	117
Slika 75. Resurs za prikaz primera lekcije <i>If naredba</i> .....	117
Slika 76. Resurs za prikaz objašnjenja lekcije <i>If naredba</i> .....	117
Slika 77. Resurs za prikaz sintaksnih pravila za učenika sa verbalnim stilom učenja .....	118
Slika 78. Resurs za prikaz sintaksnih pravila za učenika sa vizuelnim stilom učenja .....	118
Slika 79. Prikaz testa lekcije <i>For petlja</i> .....	119
Slika 80. Izveštaj o stanju kursa i trenutnim stilovima učenja .....	120
Slika 81. Pregled ponuđenih kurseva.....	120
Slika 82. Uputstvo korišćenja sistema .....	121
Slika 83. Opcije komunikacije sa mentorom.....	121
Slika 84. Pregled lekcija aktivnog kursa .....	122
Slika 85. Pregled unetih tagova .....	122
Slika 86. Stranica sistema za podešavanje osnovnih opcija .....	123
Slika 87. Odjava učenika sistema.....	123
Slika 88. Rezultati personalizacije u sistemu Protus 2.0.....	124
Slika 89. Rezultati merenja ontologije sistema Protus 2.0 .....	128

## Predgovor

Doktorska disertacija je rađena na Departmanu za matematiku i informatiku Prirodno-matematičkog fakulteta u Novom Sadu pod mentorstvom dr Mirjane Ivanović, redovnog profesora PMF-a u Novom Sadu, koja je predložila temu i rukovodila radom.

Pre svega, želim da izrazim najveću zahvalnost svojoj mentorki, dr Mirjani Ivanović, koja mi je nesebično i strpljivo prenosila svoje znanje, iskustvo i mudrost, i čija je pomoć i podrška bila bezrezervna. Zahvaljujem se na ukazanom poverenju i razumevanju, posvećenom vremenu i sjajnim idejama, i ogromnom strpljenju posebno tokom izrade i pisanja disertacije.

Neizmerno hvala članovima komisije za odbranu doktorske disertacije: dr Zoranu Budimcu (Prirodno-matematički fakultet, Novi Sad), dr Draganu Jankoviću (Elektronski fakultet, Niš) i dr Vladimiru Kurbaliji (Prirodno-matematički fakultet, Novi Sad) na posvećenom vremenu, dragocenim savetima i korisnim sugestijama koje su ovaj rad učinile boljim.

Posebnu zahvalnost dugujem svojoj kolegini dr Aleksandri Klašnja-Milićević na saradnji, podršci i prijateljskoj otvorenosti tokom svih prethodnih godina zajedničkog rada.

Nemerljivu zahvalnost dugujem mojoj mami Anki i sestri Vesni za podršku, razumevanje i ljubav koju su mi pružili, kao i svim mojim prijateljima koji su bili uz mene tokom izrade ove disertacije. Posebno bih se zahvalio Ani na ljubavi, velikom strpljenju, podsticaju i veri u uspeh. Najlepše Vam hvala!

Delovi doktorske disertacije objavljeni su u naučnim radovima na projektima finansiranim od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije i to: *Inteligentne tehnike i njihova integracija u sisteme za podršku odlučivanju sa širokim poljem primene* (projekat broj OI-174023) i *Infrastruktura za elektronski podržano učenje u Srbiji* (projekat broj III-47003).

Novi Sad, 2014

Boban Vesin

## Poglavlje I

### UVOD

Semantički veb (eng. *Semantic web*) predstavlja viziju globalne mreže kao univerzalnog medija za razmenu podataka, informacija i znanja. Semantički veb predstavlja proširenje World Wide Web-a u kojem je definisana semantika informacija i veb servisa čime se omogućuje da veb „razume“ i ispunjava potrebe ljudi i računara prilikom korišćenja različitog sadržaja (Berners-Lee et al., 2001).

Cilj semantičkog veba je transformacija nestruktuiranih podataka na internetu u eksplicitno i precizno deklarirano znanje integrisano u veb aplikacijama. Drugim rečima cilj semantičkog veba je definisanje semantike podataka na Internetu. Takođe, semantički veb je način da se pouzdano implementira duboka integrisanost veb servisa tj. da se kreira mreža od tesno povezanih servisa, koje će inteligentni agenti moći otkriti, izvršiti i automatski kombinovati (Devedžić, 2004a).

Jedna od osnovnih komponenti semantičkog veba su ontologije. Ontologija određenog domena predstavlja terminologiju (rečnik domena), skup svih bitnih koncepata u domenu, njihove klasifikacije, taksonomije, njihove veze (uključujući sve važne hijerarhije i ograničenja) i sve aksiome domena. Pri modelovanju inteligentnih aplikacija, ontologije predstavljaju važne ponovno upotrebljive gradivne blokove koje mnoge aplikacije mogu da integrišu kao već postojeće module znanja. Informacije koje su sadržane u ontologijama se uz pomoć pravila adaptacije koriste u procesu prilagođavanja sadržaja.

Razvojem semantičkog veba popularizuje se i upotreba ontologija kao formalizma za opis znanja i informacija koje se prikazuju i dele preko mreže. Eksplicitno predstavljanje komponenti sistema u formi ontologija podstiče razmenu znanja, ponovnu upotrebu informacija, komunikaciju, saradnju i razvoj bogatih i funkcionalnih sistema.

Ontologije omogućavaju formalnu i eksplicitnu specifikaciju koncepata koji se pojavljuju u određenom domenu, njihovih osobina i odnosa (Gascueña et al., 2006). Takođe su

korisne u mnogim domenima, a posebno u oblasti obrazovnih sistema budući da omogućuju ljudima i softverskim agentima da razumeju i koriste iste strukture znanja. Prednost upotrebe ontologija je što one omogućuju ponovnu upotrebu znanja, tj. nije potrebno razvijati nove ontologije ako su već odgovarajuće ontologije iz istog domena prisutne.

Iako ontologije poseduju niz osnovnih mehanizama za struktuiranje znanja izvedenih iz opisne logike (eng. *descriptive logic*) na kojoj se zasnivaju (klasifikacija, veze, identifikacija instanci klasa, itd.) njima je neophodno dodeliti niz pravila adaptacije kojima bi se automatski realizovale izmene u njima i definisali neki dodatni uslovi i odnosi između instanci klasa koje je nemoguće prikazati samo sa ontologijama. Na primer, pravila moraju biti korišćena ako bi u obrazovnom sistemu želeli da prikažemo da za savladavanje koncepta A postoji preduslov savladavanja koncepta B. Stoga, ontologije zahtevaju sistem pravila da bi se generisale nove ili koristile postojeće informacije iz njih, dok sistem pravila zahteva postojanje definicije koncepta i njihovih međusobnih odnosa. Pravila takođe dodaju izražajnost na formalizam prikaza informacija i donošenje zaključaka na osnovu podataka iz instanci klasa. Sistemi pravila se kao i ontologije prikazuju na osnovu standardne semantike i formalizma (Chi, 2009).

Sa druge strane, sistemi za elektronsko učenje u poslednjoj dekadi doživljavaju ubrzan razvoj. Takođe, elektronsko učenje ima veliki potencijal da se i dalje brzo razvija u budućnosti. Prednosti učenja preko globalne mreže su višestruke i očigledne: nezavisnost od vremena i prostora, učenici mogu da uče u sopstvenom ritmu, nastavni materijal može biti postavljen na jednom mestu a korišćen i dorađivan po celom svetu. Elektronsko učenje se stoga pokazalo kao efikasno, fleksibilno i pristupačno. Razvoj sistema za elektronsko učenje je očigledno mnogo zahtevniji i skuplji nego razvoj statičnog sistema budući da je za kvalitetniji sistem poželjno kreirati različite prikaze istog nastavnog sadržaja. Različite forme istog nastavnog materijala su neohodne za uspešnu implementaciju opcija personalizovanog učenja u sistemu. Ipak, opcije personalizacije povećavaju efikasnost sistema za elektronsko učenje, čime se opravdavaju veći inicijalni troškovi njihove izgradnje.

Jedan od najvažnijih segmenata u današnjem razvoju i upotrebi Interneta je personalizacija sadržaja i izgradnja korisničkog profila zasnovanog na ponašanju svakog pojedinačnog korisnika. Formirani profil korisnika treba da pomogne korisniku u izboru novih sadržaja i informacija koje mu se u datom trenutku predstavljaju. U cilju personalizacije procesa učenja i prilagođavanja sadržaja svakom pojedinačnom učeniku, sistemi za elektronsko učenje moraju koristiti strategije kojima će zadovoljiti potrebe učenika. Takođe, ti sistemi moraju koristiti različite tehnologije radi promene okruženja i adaptacije nastavnog materijala na osnovu potreba učenika. Proces adaptacije može biti u formi adaptacije sadržaja, procesa učenja, povratnih informacija ili navigacije.

U poslednjih nekoliko godina razvijali smo i implementirali tutorski sistem pod imenom Protus (*PRogramming TUtoring System*). Sistem se upotrebljavao i testirao u okviru predmeta *Primenjeno programiranje* za učenje osnova programskog jezika Java na Visokoj poslovnoj školi strukovnih studija u Novom Sadu (Vesin et al., 2009). Sistem Protus koristi principe identifikacije individualnih stilova učenja i sistem za generisanje preporuka u cilju personalizacije nastavnog materijala (Vesin et al. 2008; Vesin et al., 2011a).

Različiti učenici imaju različite želje, potrebe i pristupe učenju. Te razlike među učenicima su u stručnoj literaturi poznate kao individualni stilovi učenja. Individualni stil učenja se može definisati kao kombinacija informacija o ponašanju, navikama i kognitivnim karakteristikama učenika koje utiču na način na koji učenik prihvata informacije, vrši interakciju prilikom učenja i odgovara na zahteve okruženja za elektronsko učenje (Popescu et al., 2007). Stoga, veoma je važno prilagoditi sistem tim različitim stilovima učenja i nastavne sadržaje prikazati učenicima u skladu sa njihovim optimalnim stilom. Postoji više od sedamdeset pristupa prepoznatih u literaturi za identifikaciju i opis stilova učenja (Klašnja-Milićević et al. 2011b). Pri izradi tutorskog sistema Protus korišćen je jedan od najpoznatijih pristupa, *Index of Learning Styles (ILS)* (Felder & Soloman, 1996). Na osnovu njega učenici se grupišu u četiri kategorije različitih-suprotstavljenih stilova:

- *Obrada informacija*: Aktivni (eng. *Active*) i Refleksivni (eng. *Reflective*) učenici,
- *Percepcija informacija*: Osećajni (eng. *Sensitive*) i Intuitivni (eng. *Intuitive*) učenici,
- *Prijem informacija*: Vizuelni (eng. *Visual*) i Verbalni (eng. *Verbal*) učenici,
- *Razumevanje informacija*: Sekvencijalni (eng. *Sequential*) i Globalni (eng. *Global*) učenici.

Izbor optimalne navigacione sekvence kroz nastavni materijal (eng. *Resource sequencing*) predstavlja razvijenu tehnologiju u oblasti inteligentnih tutorskih sistema. Ideja je da se generiše personalizovani nastavni materijal za svakog učenika uz pomoć dinamičnog odabira optimalnih akcija, prezentacije materijala, primera, zadataka ili problema nakon svake konkretne akcije učenika tokom upotrebe tutorskog sistema. Pod optimalnom akcijom se podrazumeva operacija koja u kontekstu drugih postojećih akcija dovodi učenika bliže krajnjem cilju učenja. U većini slučajeva taj cilj predstavlja sticanje nekog nivoa znanja u optimalnom vremenskom periodu.

Osnovni cilj disertacije je definisanje opšteg modela tutorskog sistema za prikaz kurseva i nastavnog materijala iz različitih domena kao i upotreba definisanog modela za unapređenje postojećeg sistema Protus. Implementacija opcija personalizacije i integracija sistema za generisanje preporuka predstavljaju osnovna unapređenja sistema. Kao rezultat je nastala nova verzija sistema – Protus 2.0, koja se u potpunosti zasniva na tehnologijama i standardima semantičkog veba i upotrebi ontologija i pravila adaptacije.

Definisana arhitektura sistema Protus 2.0 unapređuje upotrebu ontologija, tako što je svaka komponenta sistema predstavljena posebnom ontologijom. Na taj način, biće omogućeno jasno razdvajanje uloga pojedinih komponenti tutorskog sistema, definišaće se eksplicitna komunikacija između komponenti, olakšaće se dalja izgradnja komponenti i naglasiće se prednosti upotrebe tehnologija semantičkog veba. Takođe će biti omogućena lakša ponovna upotreba komponenti sistema Protus 2.0 u budućnosti, za dalji razvoj i unapređenja sistema. Prikaz pojedinih komponenti tutorskog sistema biće dat uz pomoć tehnika konceptualnog modeliranja.

Disertacija je organizovana u četiri osnovna dela. Prvi deo predstavlja teorijski okvir istraživanja i sastoji se od tri poglavlja u kojima se prikazuju osnovne ideje i motivi izrade disertacije. U okviru poglavlja *Učenje uz upotrebu veba* prikazane su osnove elektronskog učenja, tehnike personalizacije procesa učenja i mogućnosti njihove integracije u celokupan sistem za elektronsko učenje. Sledi poglavlje *Personalizacija i adaptacija u sistemima učenja* u kojem su prikazane najpopularnije forme prilagođavanja nastavnog materijala učenicima. Treće poglavlje pod nazivom *Semantički veb* sadrži prikaz osnovnih

elemenata semantičkog veća, kao i mogućnosti primene tehnologija semantičkog veća u oblasti elektronskog učenja.

Drugi deo disertacije sadrži prikaz aktuelnih trendova iz oblasti izgradnje tutorskih sistema i sadrži poglavlje *Poređenje sistema Protus 2.0 i sličnih tutorskih sistema*. U okviru ovog poglavlja prikazana su tri karakteristična sistema i opcije personalizacije koje su u njima primenjene uz upotrebu tehnologija semantičkog veća. Ovaj deo disertacije se završava prikazom rezultata svetskih istraživanja o popularnosti tehnologija semantičkog veća.

Treći deo disertacije prikazuje detalje dizajna i implementacije sistema Protus 2.0 i sadrži dva poglavlja. Poglavlje *Izgradnja modela tutorskog sistema* prikazuje detalje opšteg modela tutorskog sistema podržanog tehnologijama semantičkog veća kao i osnovne principe izrade kurseva iz različitih domena uz pomoć tog modela. Poglavlje *Dizajn i implementacija sistema Protus 2.0* sadrži detalje o prethodnim verzijama sistema, definisanim korisničkim zahtevima nove verzije sistema, detalje o arhitekturi, kao i principi primene definisanog opšteg modela za izradu sistema Protus 2.0. Na kraju ovog poglavlja dat je prikaz sistema kao i opcije personalizacije iz ugla krajnjeg učenika.

Četvrti deo disertacije daje prikaz ocena ponuđenih ontologija uz pomoć metrika nad ontologijama, sumira rezultate rada i daje konačan zaključak.

## Poglavlje II

### UČENJE UZ UPOTREBU VEBA

Učenje preko mreže (eng. *Web-based learning*) uključuje sve aspekte i procese učenja koje koriste *World Wide Web* kao osnovnu tehnologiju i medijum za komunikaciju. Koriste se i drugi termini kao što su: onlajn učenje (eng. *online learning*), virtuelno obrazovanje (eng. *virtual education*), učenje preko Interneta (eng. *Internet-based learning*) ili obrazovanje uz pomoć kompjuterske komunikacije (eng. *education via computer-mediated communication*) (Devedžić, 2004a).

Može se reći da učenje preko mreže karakteriše (Ruttenbur & Spinkler, 2000):

- razdvojenost predavača i učenika (što je i razlika ovog tipa učenja u odnosu na tradicionalno),
- upotreba veb tehnologija za prezentovanje i distribuciju obrazovnih sadržaja,
- mogućnosti dvosmerne komunikacije koja omogućuje međusobnu komunikaciju učenika kao i komunikaciju učenika sa predavačima i drugim nastavno-administrativnim osobljem.

Od 1990. godine učenje preko mreže postala je važna grana obrazovanja. Za učenike ono nudi praktično neograničeni pristup informacijama i znanju, i takođe nudi mogućnosti prilagođavanja kurseva svakom pojedinačnom učeniku, udaljeno učenje (eng. *tele-learning*), mogućnosti međusobne saradnje učenika sa jasnim prednostima koje nudi nezavisnost od učionice i platforme (Brusilovsky, 2003).

Sa druge strane, predavači i autori nastavnih materijala mogu koristiti brojne mogućnosti onlajn kursa kao što su predavanje na daljinu (eng. *teleteaching*), autorski alati za razvoj kurseva, jeftino i efikasno skladištenje i distribucija materijala za kurs, digitalne biblioteke, itd.

Postoji određen broj važnih elemenata povezanih sa učenjem preko mreže, kao što su elektronsko učenje, učenje na daljinu i personalizovano učenje. U narednim poglavljima biće detaljnije pojašnjeni pojedini koncepti i odgovarajuće tehnologije.

## 2.1 Elektronsko učenje

Elektronsko učenje ili E-učenje (eng. *e-learning*) je interaktivno učenje gde je sadržaj koji se uči dostupan *onlajn* i gde je ponuđena automatska povratna akcija kao odgovor na učenikovu aktivnost (De Bra, 2006). U suštini, to je više učenje uz pomoć računara (eng. *computer-based training*) i davanje instrukcija uz pomoć računara (eng. *computer-aided instruction*) ali je razlika (u odnosu na tradicionalno učenje na daljinu) što je potreban Internet za pristup materijalu koji se uči i za kontrolu učenikovih aktivnosti. Elektronsko učenje nudi učenicima komunikaciju sa svojim predavačima preko Interneta. Međutim, naglasak nije na komunikaciji već na organizaciji i pristupu sadržaju koji se uči.

Po nekim definicijama, elektronsko učenje ne sadrži samo Internet kao tehničku podršku učenju nego i druge medije i resurse kao što su Intranet, audio i video diskove, satelitsko emitovanje predavanja, interaktivna televizija, bežični i mobilni uređaji, itd (Devedžić, 2006). Ipak, razvojem Internet tehnologija, samo elektronsko učenje se pretvara u Internet učenje budući da se prevashodno koriste Internet tehnologije za kreiranje, usvajanje, prenošenje i olakšanje procesa učenja (Dutta, 2006). Jedan od ciljeva elektronskog učenja je razvoj individualizovanog, razumljivog, dinamičkog sadržaja za učenje, u realnom vremenu.

Postoje dva osnovna pravca razvoja elektronskog učenja: tehnički i pedagoški. Mnogi autori stavljaju akcenat na tehnologiju. Drugi tehnologiju koriste kao sredstvo za predstavljanje sadržaja, naglašavajući potrebu za pristupom kojim se učenik stavlja u prvi plan. Za njih, elektronsko učenje je u suštini samo učenje. U pravcu razvoja koji potencira pedagoški aspekt, akcenat se stavlja na objašnjenja načina na koji ljudi uče, kako stiču sposobnosti i preuzimaju informacije, koji su njihovi stilovi učenja, itd. Tek nakon ovih primarnih aspekata, postavlja se pitanje kako elektronsko predstavljanje gradiva može biti prilagođeno učeniku.

Elektronsko učenje je obično predstavljeno kao interakcija između učenika i simuliranog elektronskog okruženja povezanog sa domenom znanja interesantnog za učenika. To elektronsko okruženje u slučaju učenja preko mreže može biti Internet (u različitim varijantama), intranet kao i kompakt diskovi. U svim tim slučajevima, nudi se efikasno okruženje za učenje kroz interaktivnu upotrebu teksta, slika, audio i video materijala, animacija i simulacija. Takođe, može biti uključena celokupna virtuelna sredina. Moguće je učiti samostalno ili u grupi i takođe samostalno određivati odgovarajući tempo napretka.

Elektronsko učenje ima niz prednosti u odnosu na tradicionalno učenje, u školskoj učionici. Prvenstveno, tempo elektronskog učenja je prilagođen učeniku. Zatim, troškovi su obično manji, ne postoje vremenska i prostorna ograničenja. Takođe, materijal koji se uči se lakše ažurira i samim tim se dobija na aktuelnosti. Dobra alternativa *običnom* elektronskom učenju je kombinovano iliti mešovito učenje (eng. *blended learning*) koje predstavlja kombinaciju tradicionalnog i elektronskog učenja. Ovaj oblik učenja podrazumeva kombinovanje tradicionalnih predavanja u učionici uz



povremeno/paralelno savladavanje gradiva uz pomoć nekih od oblika tutorskih sistema ili sistema za elektronsko učenje (Devedžić, 2006)

### **2.1.1 Organizacija materijala za elektronsko učenje**

Za uspešnost elektronskog učenja najvažnije je da instruktori organizuju materijal koji se uči na način koji odgovara interaktivnoj elektronskoj prezentaciji materijala (Vesin & Ivanović, 2004). Najgore rešenje je kopirati tradicionalno napisan materijal u fajlove i jednostavno ih prezentovati na ekranu učenicima. Nasuprot tome, sastavljanje kursa za elektronsko učenje je dug proces koji zahteva puno zalaganje instruktora i celog tima koji rade na projektu. Obrazovne institucije i univerziteti, koji nude onlajn kurseve imaju cele departmane koji se bave organizovanjem materijala za elektronsko učenje.

Postoji nekoliko jednostavnih pravila organizovanja materijala za elektronsko učenje. Oni su svi posledice opštih pedagoških pravila kao i interakcije čovek-računar:

- Bitno je tačno definisati ciljnog korisnika (učenici i njihov nivo znanja) i ciljeve kursa (tj. koja znanja se očekuje da učenik stekne nakon završetka kursa). U slučajevima kada se kurs izvodi preko mreže, organizacija materijala na serverskoj strani je samo deo problema - potrebno je imati u vidu i hardver koji učenik koristi, kao i brzinu protoka informacija da bi se osigurao prenos u realnom vremenu.
- Potrebno je podeliti materijal na module kao što su poglavlja ili lekcije, čime bi se omogućilo učenicima da steknu uvid u opštu strukturu materijala i steknu sliku o ciljevima kursa kao i da lakše prate detalje u poglavljima i lekcijama.

Postoje korisnički alati koji pomažu instruktorima da pripreme materijal po prethodno navedenim uputstvima (Devedžić, 2004b).

## **2.2 Popularnost elektronskog učenja**

Osnovni cilj koji se mora postići razvojem sistema za elektronsko učenje je odgovarajući nivo kvaliteta usluga (eng. *quality-of-service*) koje će sistem ponuditi (Devedžić, 2006). U sistemima za učenje preko mreže čiji je najvažniji korisnik učenik, kvalitet usluga se meri kvalitetom nastavnog materijala, kvalitetom implementirane pedagogije sistema i kvalitetom tehnološkog okvira (Vouk et al., 1999).

*Kvalitet nastavnog materijala.* Logično je da će učenici želeti da češće pristupe nastavnom materijalu visokog kvaliteta i da će steći bolje razumevanje tema pokrivenih takvim materijalom. U cilju obezbeđivanja visoke prihvaćenosti nastavnog materijala od strane različitih učenika, neophodna je neprekidna provera materijala i insistiranje na njegovom čestom ažuriranju i dopunjavanju na osnovu rezultata tih provera.

*Kvalitet pedagogije sistema.* Dobra pedagogija sistema znači da on ima kvalitetnu adaptaciju potrebama, ciljevima i stilovima učenja kako pojedinačnih učenika tako i različitih grupa učenika. Cilj je da se u sistemu obezbedi efikasan način za ocenjivanje rada učenika i ponudi personalizovani nastavak kursa u smislu odgovarajućeg tempa i tipa prezentovanog materijala. Ova personalizacija kursa treba da se realizuje na osnovu praćenja učenikove interakcije sa sistemom i procene napretka učenika.

*Kvalitet tehnološkog okvira.* Pored velikog značaja sadržaja i pedagogije zarad efikasnosti sistema, takođe je važan kvalitet upotrebljenih tehnologija. Atraktivna tehnologija često

zna da bude osnovni uzrok prihvatanja sistema i obrnuto, loša tehnologija i loše performanse sistema mogu isfrustrirati učenika i demotivirati ga.

Neke od mera kvaliteta usluga sistema se odnose na kvalitet mreže koja se koristi i krajnje su objektivne. Na primer, brzina protoka informacija, brzina odziva sistema, verovatnoća gubitka podataka, kapacitet broja korisnika, itd. Druge mere se odnose na neke subjektivne faktore, npr. pristupačnost sistema, pouzdanost izvršavanja, skalabilnost, efikasnost, itd.

### **2.2.1 Inteligencija i adaptivnost sistema za elektronsko učenje**

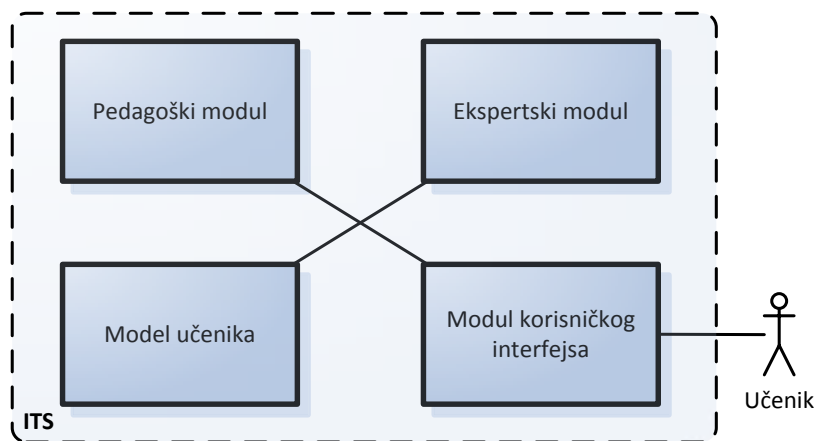
Brojni primeri inteligentnih tutorskih sistema prilagođenih radu preko mreže razvijeni su u svetu poslednjih godina kao posledica potrebe ubrzanja procesa učenja. Ovakvi sistemi su uglavnom platformski nezavisni i omogućuju pristup nastavnom materijalu sa različitih lokacija (Romero & Ventura 2006). U njima se sakuplja velika količina informacija o korisničkim sesijama čime se omogućuje analiza ponašanja učenika. U sistemima se generiše i velika količina različitih lekcija i nastavnih resursa te je takvom bazom podataka teško ručno manipulirati i prilagođavati nastavni materijal potrebama pojedinačnog učenika (Diković, 2009). Dva efikasna načina poboljšanja kvaliteta usluga sistema su: upotreba inteligencije u sistemu i adaptivnost ka interesima pojedinačnih učenika (Devedžić, 2006). Uvođenje inteligencije u sisteme za elektronsko učenje predstavlja formiranje nove kategorije sistema za elektronsko učenje i implementaciju Intelligentnih tutorskih sistema (eng. *Intelligent tutoring systems* - ITS). U inteligentnim tutorskim sistemima primenjuju se razne tehnike u cilju postizanja personalizacije. Poslednjih godina u svetu, popularna tehnika prilagođavanja nastavnog materijala je upotreba sistema za generisanje preporuka. Razni obrazovni sistemi generišu preporuke učeniku i prikazuju mu nastavni materijal u formi koja je učeniku odgovarajuća, koja će mu držati pažnju ili koju će lakše savladati.

Sistemi za generisanje preporuka koriste mišljenja zajednice korisnika kako bi pomogli pojedinačnom korisniku kako bi što efikasnije identifikovao sadržaj koji ga interesuje u velikoj količini materijala (Resnick et al., 1994). Takvi sistemi su postali moćan alat u mnogim oblastima elektronskog poslovanja, digitalnih biblioteka i pretrage baza podataka (Schafer, 2005). Neki sistemi za generisanje preporuka se takođe koriste u sistemima za elektronsko učenje u cilju preporuke učeniku koju lekciju bi trebalo narednu da uči (Kristofic, 2005) ili za preporuku koji bi nastavni materijal eventualno mogao da doprinese postizanju prethodno definisanog cilja učenja (Farzan & Brusilovsky, 2006). Sistemi za generisanje preporuka imaju drugačije zahteve u sistemima za elektronsko učenje u odnosu na druge oblasti. Tako je u obrazovnim sistemima potrebno uzeti u obzir pedagoške aspekte učenika i potrebu da se nastavni materijal preporučuje u redosledu koje nalaže sam proces učenja. U ovoj disertaciji prikazana je konkretna integracija sistema za generisanje preporuka učeniku u sistem za elektronsko učenje koji uzima u obzir profil učenika i istoriju njihovog pristupa resursima sistema uz pomoć tehnika *data mining*-a.

### **2.2.2 Intelligentni tutorški sistemi**

Intelligentni tutorški sistemi ili inteligentni obrazovni sistemi koriste metode i tehnike veštačke inteligencije u cilju poboljšanja kvaliteta procesa predavanja i učenja uz pomoć računara.

Na slici 1 prikazana je tradicionalna arhitektura inteligentnih tutorskih sistema (Padayachee, 2000). Domen znanja ili sadržaj instrukcija inteligentnih tutorskih sistema se nalazi u *ekspertskom modulu*, dok različite strategije učenja iz *pedagoškog modula* usmeravaju tok korisničkih sesija učenika. Svrha *modela učenika* je modeliranje osobina učenika i njegovog napretka u savladavanju nastavnog materijala. Cilj modeliranja učenika je dinamička adaptacija procesa i dinamička dodela instrukcija učeniku. Inteligentni tutorski sistemi podržavaju samostalno učenje, gde učenik prima instrukcije od veštačkog tutora (prethodna tri spomenuta dela sistema) i pritom se sva komunikacija obavlja preko *modula korisničkog interfejsa*. Postoje i sistemi koji podržavaju saradnju između učenika tako što dozvoljavaju većem broju učenika da uče kao grupa kroz interakciju sa sistemom. Svi inteligentni tutorski sistemi koriste različite reprezentacije znanja i tehnike rezonovanja iz domena veštačke inteligencije. Postoje alati za kreiranje inteligentnih tutorskih sistema koji predstavljaju integrisana softverska okruženja čija je osnovna uloga razvoj inteligentnih tutorskih sistema (Devedžić, 2004a).



**Slika 1. Tradicionalna arhitektura inteligentnih tutorskih sistema**

Model učenika je obično slojeviti model, što znači da je predstavljen skupom numeričkih vrednosti od kojih svaka vrednost prikazuje nivo znanja ili osposobljenosti učenika za svaku pojedinačnu temu ili koncept iz domena znanja. Numeričke vrednosti predstavljaju nivo znanja. Vrednosti za konkretnog učenika se neprekidno ažuriraju tokom učenja i promene u učenikovom znanju i učenju se beleže u modelu učenika.

U cilju prilagođavanja sistema učeniku i povećanja efikasnosti učenja, što su i krajnji ciljevi inteligentnih tutorskih sistema, važno je naglasiti usku povezanost modela učenika i personalizacije procesa učenja. U modelu učenika se beleže svi neophodni parametri o pojedinačnom učeniku, kao što su njegovi ciljevi učenja, prethodno postignuti rezultati na testovima, stil učenja, trenutni nivo znanja i ostale osobine. Informacije sadržane u ovom modelu omogućuju definisanje daljih aktivnosti personalizacije nastavnog materijala. Inicijalizacija narednih korisničkih sesija se vrši na osnovu tih informacija (npr. trenutna lekcija, savladani/nesavladani materijal, stil učenja, itd).

## **Poglavlje III**

### **PERSONALIZACIJA I ADAPTACIJA U SISTEMIMA UČENJA**

Personalizacija obrazovnog sistema predstavlja prilagođavanje raspoloživog sadržaja i formi nastavnog materijala za prikaz učeniku u procesu učenja a na osnovu podataka iz modela učenika. Postoje različiti tipovi personalizacije u pogledu forme prilagođavanja teksta, prikazivanja odgovarajućih stranica ili primena različitih formi lekcija. Sadržaj se može prilagođavati pružanjem dodatnog materijala, naglašavanjem bitnih informacija ili uporednim prikazom različitih vrsta i tipova objašnjenja (Devedžić, 2006; Klašnja-Milićević et al. 2011b).

U principu postoje različite kategorije korisnika sistema za elektronsko učenje. Oni se razlikuju po starosnom dobu, individualnim mogućnostima, kulturološkim karakteristikama, zanimanju, itd. Neki od njih su učenici, neki od njih su studenti a neki od njih zaposleni u radnim organizacijama. Neki od korisnika su u svojim kompanijama zaduženi za obuku zaposlenih pa ovakve sisteme koriste za povećanje efikasnosti zaposlenih. Takođe, često obrazovne sadržaje na internetu pretražuju zaposleni u vladinim organizacijama, istraživači ili administrativni radnici. Bez obzira na njihove razlike oni su svi zainteresovani za interesantno i efikasno učenje.

Zadovoljstvo korisnika i efikasnost učenja u mnogome će zavisi od sadržaja koji se uči, načina na koji se prezentuje, efikasnosti, aktuelnosti, dostupnosti tog sadržaja, itd. Budući da su svi korisnici različiti, takođe različite personalizacije moraju biti uključene prilikom razvoja sistema za elektronsko učenje u cilju podrške pojedinačnim, jedinstvenim karakteristikama korisnika. Personalizacija obično podrazumeva uzimanje u obzir različitih nivoa znanja korisnika i primena raznih vidova pomoći i dodatnih objašnjenja, različitih putanja kroz kurs, različitih principa i tehnika adaptivnih hipermedija, itd. Pored toga, sistem za elektronsko učenje mora obezbediti odgovarajuću sigurnost podataka, prikaz različitih izveštaja o rezultatima učenja, ocenama, rangiranjima, kao i razne statistike učenja.

Personalizacija, njen kvalitet i efikasnost u procesu učenja zavisi od kvaliteta integrisanog modela učenika i kao i samog načina modeliranja učenika. Modeliranje učenika obuhvata širok spektar elemenata počev od skupa aktivnosti, procesa, znanja, struktura podataka do odgovarajućih alata i tehnologija za modeliranje učenikovog prethodnog znanja, sposobnosti, potreba, tempa savladavanja gradiva, itd. Cilj modeliranja je da se u sesijama učenja pruži dinamička adaptacija procesa davanja instrukcija svakom pojedinačnom učeniku.

### **3.1 Osnove personalizacije**

Personalizacija igra kompleksnu ulogu u sistemima za elektronsko učenje budući da mora predstavljati ogledalo različitosti učenika. Učenici se razlikuju po nivou znanja, napretku u učenju, ličnim kapacitetima, itd. Često karakteristike učenika nisu statične, čak naprotiv, većina tih karakteristika se konstantno menja tokom vremena i dobija različite vrednosti tokom različitih sesija. Iz tog razloga personalizacija zahteva ne samo specifičnu reprezentaciju struktura i tehnika već i posebne autorske zadatke i strategije u cilju reprezentacije modela učenika.

Postoji više različitih tipova personalizacije kao i različitih pristupa za postizanje personalizacije sistema za elektronsko učenje.

#### **3.1.1 Tipovi personalizacije**

Istog trenutka kad započne interakcija učenika i sistema za učenje, mora početi i adaptacija sistema na način na koji učenik želi da komunicira i organizuje informacije (Romero et al., 2007). Personalizacijom se sistem adaptira primenom efikasnih algoritama adaptacije i odgovarajućih struktura podataka da bi se predstavile karakteristike učenika i načini na koji on rešava probleme. Personalizacija podrazumeva da sistem gradi i koristi model ponašanja učenika tokom međusobne interakcije sistema i učenika tokom sesija. Da bi se to postiglo, sistem mora da ažurira celokupnu istoriju svih zahteva učenika, objekata učenja koji su predstavljeni učeniku, specifičnih pojava konteksta u kojem je znanje prikazivano kao i istoriju učenikovih odgovora.

Ukratko, osnovne uloge personalizacije u sistemima za učenje su (Devedžić et al., 2006):

- Personalizovani sistemi za učenje omogućuju učenje u formi jedan predavač – jedan učenik i više predavača – jedan učenik, za razliku od običnih sistema za učenje gde je odnos uvek jedan predavač – više učenika.
- Personalizovani sistemi ne poseduju nikakva ograničenja u smislu vremena, lokacije, itd, dok u običnim sistemima najčešće postoje ograničenja prethodno utvrđenog rasporeda.
- Personalizovani sistemi prepoznaju različite karakteristike i zahteve učenika u smislu različitih stilova učenja, interesovanja, načina učenja i prilagođavaju davanje instrukcija na osnovu njih. Tradicionalni sistemi su obično prilagođeni prosečnom učeniku.
- Personalizovani sistemi formiraju instrukcije da bi se zadovoljili zahtevi učenika. U tradicionalnim sistemima predavač utvrđuje raspored jedinica učenja, selekciju i raspored nastavnog materijala koji se uči.

Personalizacija može podrazumevati različite aktivnosti u sistemu od adaptacije sadržaja materijala do prilagođavanja korisničkog interfejsa. Predloženo je pet tipova personalizacije, na pet različitih nivoa apstrakcije i preciznosti (Devedžić, 2006):

- **Personalizacija prepoznavanja imena** (eng. *name-recognized personalization*). Ovaj oblik personalizacije uključuje prepoznavanje korisnika i prikaz podataka o njemu na ekranu. Na primer, prikazivanje imena učenika kao deo nastavnog materijala, problema koji učenik rešava ili rezultata koji je postigao. Ova je strategija jednostavna i laka za implementaciju a ipak vredna budući da se mnogi ljudi osećaju priznatim kao pojedinci kada ugledaju svoje ime na ekranu.
- **Personalizacija uz lični opis** (eng. *self-described personalization*). Prilikom registrovanja korisnik unosi podatke o sebi, koji u kasnijem procesu personalizacije mogu biti od koristi za prilagođavanja sadržaja nastavnog materijala i korisničkog interfejsa. Uz pomoć upitnika, početnog testiranja ili formulara za registraciju, učenici mogu opisati svoje potrebe i atribute kao i da identifikuju svoje poreklo i prethodno iskustvo. Na taj način se formira početni model učenika kao osnova za početak učenja davanja instrukcija u prvoj sesiji. Na primer, ako se učenik pri početnom unosu podataka izjasni da poseduje određeni početni nivo znanja sistem može da preskoči neke od inicijalnih lekcija i kurs započne od obrade nepoznatog gradiva.
- **Segmentirana personalizacija** (eng. *segmented personalization*). Na osnovu profila učenika, sistem formira kategorije korisnika na osnovu njihovih zajedničkih osobina, znanja ili navika. Ovaj tip personalizacije uključuje grupisanje učenika u manje grupe, koje su jednostavnije za identifikaciju i upravljanje, na osnovu nekih atributa (razred, departman, titula), demografije ili rezultata anketiranja. Delovi instrukcija su tada namenjeni pojedinim grupama i primenjuju se identično za sve pripadnike te grupe.
- **Kognitivna personalizacija** (eng. *cognitive-based personalization*). Formiranje i predstavljanje sadržaja i instrukcija se vrši namenski za pojedine tipove učenika, definisano na osnovu informacija o njihovim kognitivnim procesima, strategijama, sposobnostima, i željama pri učenju. Ovakva personalizacija uključuje, na primer, učenikov izbor između audio i tekstualnog predstavljanja sadržaja, ili izbor između linearnog i nelinearnog učenja. Takođe se prepoznaju učenikove sposobnosti memorisanja informacija i mogućnosti induktivnog rezonovanja u okviru identifikovanih stilova učenja. Ovaj tip personalizacije je teži za implementaciju od prethodnih budući da zahteva sakupljanje podataka, praćenje učenikovih aktivnosti, poređenje sa ponašanjima drugih učenika i predviđanje šta bi učenik želeo da radi ili vidi sledeće. Ipak, kognitivna personalizacija predstavlja moćan i koristan vid personalizacije, budući da se učenje bazirano na njoj pokazalo kao prirodnije i efikasnije.
- **Potpuna personalizacija** (eng. *whole-person personalization*). Ova personalizacija predstavlja kombinaciju kognitivne personalizacije i skupa utemeljenih psiholoških izvora koji utiču na razlike u učenju i učinku učenika. Ovim tipom personalizacije se vrše inteligentne izmene podataka o učeniku tokom njegovog učenja i dinamičko ažuriranje učenikovog modela. Česta tema najnovijih istraživanja je utvrđivanje veza između pojedinačnih karakteristika ličnosti, učinka i sposobnosti učenika (Kim & Schniederjans, 2004).

Analizom trendova u modernim sistemima za elektronsko učenje moguće je primetiti da tipovi personalizacije koji se najčešće sreću u današnjim sistemima za elektronsko učenje su (Klašnja-Milićević, 2011b):

- **Identifikacija stilova učenja** (eng. *Learning styles identification*). Personalizacija u sistemu se vrši na osnovu identifikovanih stilova učenja svakog korisnika sistema.
- **Upotreba sistema za generisanje preporuka** (eng. *Recommendation systems*). Ovi sistemi se koriste u cilju preporuke adekvatnog nastavnog materijala učeniku ili odabira optimalne putanje kroz nastavni materijal.
- **Prilagođavanje linkova** (eng. *Link adaptation*). Ovaj vid personalizacije uključuje prilagođavanje korisničkog interfejsa i linkova sa ciljem prikaza koji je materijal nov, savladan ili samo posećen.

Prva implementirana verzija sistema Protus nudi elemente svih prethodnih tipova personalizacije te će o njima biti više reči u nastavku. Cilj disertacije je implementacija opcija personalizacije u novoj verziji sistema Protus 2.0 uz pomoć tehnika semantičkog veća.

### 3.2 Identifikacija stilova učenja

Jasno je da različiti učenici imaju različite potrebe, želje i pristupe procesu učenja. Psiholozi te različitosti nazivaju različiti individualni stilovi učenja. Stoga je jako važno sisteme za elektronsko učenje prilagoditi učenikovom stilu učenja sa ciljem postizanja efikasnijeg savladavanja materijala. Stilovi učenja se definišu kao jedinstveni način na koji se učenici koncentrišu, procesiraju, prihvataju i pamte nove i teške informacije (Dunn et al., 1984). Stilovi učenja predstavljaju pojedinačne šablone učenja koji variraju od osobe do osobe. Neophodno je utvrditi koje akcije će podstaći bolju koncentraciju učenika, kako je održati i kakav odgovor dati na postupke učenika da bi podstakli efikasnije savladavanje gradiva.

Termin *stilovi učenja* predstavlja ideju utvrđivanja razlika između učenika na osnovu načina učenja i prezentacije nastavnog materijala koji daju najbolje efekte, (Pashler et al., 2009). Zagovornici identifikacije stilova učenja tvrde da se optimalno učenje postigne utvrđivanjem kategorije stila učenja i prilagođavanjem zadavanja instrukcija učenicima na osnovu utvrđenih stilova. Mnogi stilovi učenja se pominju u literaturi od kojih su najpopularniji model stilova učenja Felder & Silverman (1988), Honey & Mumford (1982), Kolb (1984) i Pask (1976).

Prilikom izrade prve verzije sistema Protus (Klašnja-Milićević, 2011b), korišćen je pristup identifikaciji stilova učenja na osnovu modela Felder-Silverman (eng. *Felder-Silverman Learning Styles Model - FLSM*) (Felder & Silverman, 1988). Ovaj model opisuje stilove učenja kroz kategorizaciju učenika u okviru četiri dimenzije: Obrada informacija (eng. *Information Processing*), percepcija informacija (eng. *Information Perception*), prijem informacija (eng. *Information Reception*) i razumevanje informacija (eng. *Information Understanding*).

U disertaciji će biti predstavljeni principi identifikacije stilova učenja implementirani u novoj verziji sistema Protus.

### 3.2.1 Indeks stilova učenja

U prvoj verziji sistema Protus (Klašnja-Milićević, 2011b) za početnu identifikaciju individualnih stilova učenja je korišćen postupak pod imenom Indeks stilova učenja (Felder & Soloman, 1996). Ovaj upitnik sadrži 44 pitanja sa ponuđenim odgovorima kojima se procenjuju varijacije u stilovima učenja u okviru četiri dimenzije (domena). Dimenzije su: Obrada informacija, Percepcija informacija, Prijem informacija i Razumevanje informacija. U okviru svake od dimenzija postoje dve kategorije učenika (tabela 1):

- Obrada informacija: Aktivni i Refleksivni učenici,
- Percepcija informacija: Osećajni i Intuitivni učenici,
- Prijem informacija: Vizuelni i Verbalni učenici,
- Razumevanje informacija: Sekvencijalni i Globalni učenici.

Tabela 1. Karakteristike identifikacije stilova učenja na osnovu modela Felder-a i Soloman-a

Aktivni	Refleksivni
Rad u grupama	Samostalan rad
Želja da isprobaju nov materijal (Pitanja, rasprava, objašnjenja)	Želja da razmisle detaljno o problemu
Praktičari	Teoretičari
Osećajni	Intuitivni
Strpljivi sa detaljima	Više zainteresovani za opšte preglede i šire znanje (dosadno im je detaljisanje)
Upotreba standardnih metoda	Inovacije
Činjenice i eksperimenti	Percepcija, principi i teorije
Vizuelni	Verbalni
Radije prihvataju nastavni material u vidu slika, dijagrama i grafika (vizuelne forme)	Radije prihvataju tekstualni nastavni material
Globalni	Sekvencijalni
Želja da steknu prvo globalnu sliku	Želja da postepeno procesiraju informacije
Prihvataju informacije u nepovezanim delovima i razumevaju gradivo u velikim koracima bez udubljanja u detalje	Prihvataju i razumeju informacije linearno i postepeno ali bez uvida u opštu sliku

#### Domen obrade informacija: aktivni i refleksivni učenici

*Domen obrade informacija* obuhvata učenike sa **refleksivnim** stilom učenja kojima više odgovara da obrađuju brojne primere i **aktiviste** kojima više odgovara da uče kroz obavljanje pojedinih aktivnosti. Aktivni učenici najbolje uče kroz različite aktivnosti u vidu diskusija, direktne primene objašnjenih koncepata ili tako što objašnjavaju gradivo drugima. Refleksivni učenici su osobe koje radije sakupljaju i analiziraju informacije pre nego što pređu na aktivnosti. Oni će radije pregledati mišljenja i tumačenja drugih ljudi nego da sami učestvuju u rešavanju problema. U sistemu Protus, učenicima sa aktivnim stilom učenja se prvo prikazuju resursi koji iniciraju aktivnosti a potom se daju primeri, objašnjenja i teorija (Klašnja-Milićević et al., 2011b). Za učenike sa refleksivnim stilom učenja raspored predstavljanja resursa je drugačiji, prvo se prikazuju primeri, opisi i teorija a tek na kraju se od njih očekuje da učestvuju u aktivnostima.



### **Domen percepcije informacija: osećajni i intuitivni učenici**

U okviru *Domena percepcije informacija*, učenici iz kategorije **osećajnih** su strpljivi sa detaljima, lako memorišu činjenice i uspešno obavljaju praktične vežbe. Sa druge strane, **intuitivni** učenici su uspešniji pri razumevanju novih koncepata i lakše prihvataju gradivo predstavljeno preko apstraktnih i matematičkih formula. Učenici sa osećajnim stilom učenja radije rešavaju probleme jasno definisanim metodama i nisu skloni komplikovanju. Sa druge strane, intuitivni učenici vole inovacije i nisu skloni ponavljanju. Osećajni učenici su praktičniji i pažljiviji nego intuitivni učenici koji brže rade i skloni su inovacijama.

Pretpostavlja se da će osećajni učenici biti zainteresovani za dodatni materijal, te im se linkovi ka tim resursima preporučuju tokom kursa (Klašnja-Milićević et al., 2011b). Prezentacija lekcija za osećajne učenike će činiti apstraktni materijal, formule i definicija osnovnih koncepata. Odgovarajuća objašnjenja im se prikazuju u vidu blok dijagrama i tačnih sintaksnih pravila

### **Domen prijema informacija: vizuelni i verbalni učenici**

Učenici iz kategorije **vizuelnih** učenika u okviru *Domena prijema informacija* najbolje pamti informacije koje su prikazane u obliku slika, dijagrama, grafika ili video demonstracija. **Verbalni** učenici u okviru istog domena lakše zapamte informacije uz pomoć tekstualnih opisa. U sistemu Protus, učenicima iz obe kategorije domena prijema informacija se prikazuje odgovarajući materijal sa grafičkim ili tekstualnim opisima lekcija (Klašnja-Milićević et al., 2011b).

### **Domen razumevanja informacija: sekvencijalni i globalni učenici**

Učenici u okviru *Domena razumevanja informacija* se takođe razvrstavaju u dve kategorije: na učenike sa **sekvencijalnim** i **globalnim** stilom učenja. Sekvencijalni učenici teže praćenju logičke putanje kroz nastavni materijal prilikom potrage za rešenjima problema. Sa druge strane, učenici sa globalnim stilom učenja su spremni da brže rešavaju kompleksne probleme i razumevaju nastavni materijal ako prethodno steknu globalnu sliku o gradivu. Učenici sa sekvencijalnim stilom učenja radije prolaze kroz kurs linearno, gde svaki korak logično prati prethodni, dok globalni često preskaču materijal i ubrzano prelaze na složenije koncepte. U zavisnosti od kategorije kojoj učenici pripadaju u okviru domena razumevanja u sistemu Protus njima se prikazuju ili skrivaju elementi korisničkog interfejsa koji omogućuju globalni pregled nastavnog materijala kao i linkovi za linearno kretanje kroz kurs (Klašnja-Milićević et al., 2011b).

## **3.3 Sistemi za generisanje preporuka**

Sistemi za generisanje preporuka koji omogućavaju korisnicima lakši pristup resursima i odabir opcija iz mnoštva mogućnosti, prisutni su u različitim raspoloživim softverskim sistemima. Softverski sistemi iz brojnih oblasti (filmska industrija, elektronska trgovina, itd.) imaju integrisane sisteme za generisanje preporuka. Na taj način, sistemi kao što su amazon.com, netflix.com, drugstore.com ili ebay.com (Linden et al., 2003; Schafer et al., 1999) koriste sisteme za generisanje preporuka da bi usmerili pažnju korisnika ka odgovarajućim proizvodima. Osnovna namena ovih sistema da izvrše preselekciju informacija za koje bi korisnik mogao biti zainteresovan (Adomavicius & Tuzhilin, 2005).

Najvažniji cilj generisanja preporuka u sistemima elektronskog poslovanja je da se ponude korisniku informacije koje bi mu pomogle prilikom izbora proizvoda za kupovinu. Postojeći sistemi za generisanje preporuka u sistemima za elektronsko poslovanje inspirisali su mnoge da dizajniraju i implementiraju posebne komponente za generisanje preporuka u sistemima za elektronsko učenje (Fertalj et al., 2010).

Sistemi za generisanje preporuka prepoznaju potrebe veb korisnika i nude im preporuke akcija ili resursa za dalju upotrebu sistema (Badica et al., 2011). Ovi sistemi stižu veliku popularnost u sajtovima za elektronsku trgovinu u cilju generisanja preporuka za onlajn kupovinu (Klašnja-Milićević et al. 2011a). Sistemi za generisanje preporuka se koriste i u sistemima za elektronsko učenje u cilju preporuke akcija, resursa ili jednostavno linkova ka preporučenom nastavnom materijalu. Sistemi za generisanje preporuka se mogu zasnivati na više tehnika od kojih su najčešće:

- klasterovanje podataka (eng. *Data clustering*),
- kolaborativno filtriranje (eng. *Collaborative filtering*) i
- analiza pravila pridruživanja (eng. *Association rule mining*) (Klašnja-Milićević et al. 2011b).

U sistemima za učenje potrebno je uzeti u obzir da učenici imaju specifične potrebe i karakteristike kao što su različiti nivoi znanja, stilovi učenja, motivacije, prethodno iskustvo, kognitivne sposobnosti kao i da učenici žele da postignu odgovarajući napredak u željenom vremenu. Te specifične osobine učenika nalažu da se i različiti učenici tretiraju na različit način.

Prva verzija sistema Protus koristi metode klasterovanja podataka i izbora optimalne putanje kretanja kroz nastavni materijal u cilju generisanja preporuka. Cilj disertacije je implementacija nove verzije sistema Protus koja će primeniti sisteme za generisanje preporuka uz korišćenje tehnologija semantičkog veba.

### **3.3.1 Klasterovanje podataka i korisnika**

Postoji niz situacija i okolnosti u kojima je zgodno korisnike obrazovnog sistema podeliti u konzistentne i srodne grupe. Ovaj postupak je poznat kao klasterovanje. Učenici se grupišu u manje, kompaktnije klustere (kategorije) na osnovu nekih zajedničkih osobina (nivo znanja, stilovi učenja, itd) ili sličnih potreba ili želja za posebnim načinima prezentacije materijala. U tom slučaju se nastavni materijal takođe grupiše u klustere i prezentuje na isti ili sličan način svim članovima jednog klastera učenika. Klasteri u koje se kategoriše nastavni materijal predstavljaju različite forme materijala prilagođenih različitim kategorijama učenika. Klasterizacija učenika se može postići različitim upitnicima (koje učenik popunjava na početku pohađanja kursa ili u njegovom toku) i/ili praćenjem akcija i napretka učenika. U sistemu Protus, klasterovanje se vrši na osnovu identifikovanog stila učenja svakog učenika i na osnovu toga se i generiše različit nastavni materijal. Materijal je grupisan tako da pojedine kategorije materijala podržavaju odgovarajući stil učenja.

### **3.3.2 Izbor optimalne putanje kretanja kroz nastavni materijal**

Kada se definiše struktura nekog kursa, autori kursa definišu neki logičan raspored nastavnog materijala koji će se prikazivati učeniku i pretpostavljaju da će učenici da koriste taj unapred definisani raspored. Sa druge strane, učenici često savladavaju

gradivo po nekom svom jedinstvenom rasporedu i na taj način generišu specifične sekvence svojih aktivnosti. Često niz aktivnosti koji definiše autor kursa nije optimalan za tog učenika i razlikuje se od niza koji bi sam učenik odabrao. Svi ti nizovi različitih aktivnosti koje učenici naprave se prate i zapisuju, sortiraju po uspehu učenika i generišu preporuke za naredne učenike (Klašnja-Milićević et al. 2011a). Stoga, automatsko generisanje preporuka u prvoj verziji sistema Protus nije vezano za predefinisani niz nastavnog materijala nego delimično zavisi i od aktivnosti drugih učenika. Ova tehnika se koristi u cilju izbora optimalne putanje kretanja kroz nastavni materijal (eng. *Resource sequencing*) i generisanja preporuka linkova ka nekim resursima kojim bi učenici efikasnije savladavali gradivo. Protus prati šablone ponašanja učenika, beleži resurse koje posećuju i poredi rezultate testiranja učenika. Na osnovu tih podataka Protus generiše personalizovane preporuke nastavnog materijala odgovarajućem učeniku. Pre nego što izgradi dovoljnu bazu podataka navigacionih sekvenci i njihovih rejtinga, Protus prikazuje nastavni materijal u podrazumevajućem početnom rasporedu.

### **3.4 Personalizacija prilagođavanjem linkova**

Nastavni materijal u obrazovnim sistemima učenik najčešće ne posećuje linearno nego ima mogućnosti pristupa različitom nastavnom materijalu uz pomoć brojnih linkova ka drugim lekcijama ili nastavnim jedinicama. Ti linkovi mogu biti u obliku navigacionih elemenata (dugmadi) koji se nalaze na veb stranici ili u obliku posebno naglašenog dela teksta u formi hiperlinka.

U modernom učenju preko mreže izbegava se stvaranje statičkog materijala za učenje koji se prelazi u nekom određenom rasporedu, zbog velike količine međusobne zavisnosti i uslovnih veza između stranica materijala (De Bra, 2004). Stoga, metode i tehnike adaptivne hipermedije (eng. *adaptive hypermedia*) omogućuju informisanje učenika o tome da pojedini linkovi:

- vode ka materijalu za koje nisu spremni,
- predlažu posetu pojedinim korisnim stranicama ili
- nude dodatna objašnjenja.

Adaptivni obrazovni hipermedijalni sistemi (eng. *adaptive educational hypermedia systems*) primenjuju različite oblike modela učenika u cilju prilagođavanja sadržaja i linkova ka hipermedijalnim stranicama kursa svakom pojedinačnom učeniku (Henze, 2005). Adaptivni obrazovni hipermedijalni sistemi omogućuju personalizovano učenje upotrebom tehnologije za neprekidno merenje učenikovog znanja i napretka. Sakupljeni podaci o učeniku se koriste za dalje prilagođavanje prikaza nastavnog materijala potrebama, tempu rada, željama i ciljevima učenika.

#### **3.4.1 Adaptivna hipermedija**

Hipermedijalni sistemi predstavljaju kombinaciju teksta, hiperteksta, hiperlinkova, audio i video materijala u cilju formiranja nelinearnog informacionog medija. Adaptivna hipermedija (eng. *adaptive hypermedia*) predstavlja tehniku prilagođavanja (prikazivanja, naglašavanja ili sakrivanja) hiperlinkova sa ciljem odabira odgovarajućeg sadržaja za korisnika. Postoje dva osnovna razloga upotrebe adaptivne hipermedije: izbegavanje problema koji se javljaju kada se materijal čita mimo predviđenog rasporeda i bolja adaptacija individualnim razlikama između učenika (De Bra, 2006).

Adaptivni hipermedijalni sistemi mogu da sortiraju, naglase ili sakriju linkove ka pojedinim veb stranicama u cilju odabira sadržaja koji će biti prikazan i preporuke učenicima koje resurse mogu da posete na osnovu analize njihovih ličnih karakteristika (ciljeva, potreba ili stepena znanja). Učenici takođe mogu na taj način dobijati informacije o važnosti i značaju pojedinih elemenata nastavnog materijala.

Adaptivni hipermedijalni sistemi spajaju hipermediju sa tehnikama za modeliranje korisnika i mogu se upotrebljavati u različitim oblastima primene, mada među njima dominira obrazovanje (De Bra, 2006). Cilj adaptivne hipermedije je da se premoste problemi prezentacije istog sadržaja različitim učenicima. Pristup uz pomoć adaptivne hipermedije predstavlja sakupljanje podataka o svakom korisniku i adaptaciju interakcije sa njim na osnovu tih informacija.

Adaptivna hipermedija nudi dve opšte kategorije adaptacije:

- Adaptacija sadržaja (eng. *content adaptation*) – predstavlja različitu prezentaciju sadržaja prema modulu znanja iz domena (koncepti, njihove veze, informacije o preduslovima predstavljanja materijala...) i informacija iz modela učenika (De Bra, 2006).
- Adaptacija veza (eng. *link adaptation*) ili adaptivna navigacija – sistem modifikuje dostupnost i/ili pojavu svih veza (linkova) koje postoje na *veb* strani. Na taj način učeniku se prikazuje da li te veze vode ka novim interesantnim informacijama (npr. link se naglasi posebnom bojom fonta), novim informacijama za koje učenik još nije spreman (npr. link se ukloni ili deaktivira) ili ka stranici koja ne donosi novo znanje (Romero et al., 2007).

Dva najpopularnija oblika adaptivne navigacije su:

- naglašavanje linkova (eng. *link annotation*) - predstavlja prikaz dodatnih informacija o stranici do koje vodi link (uz pomoć promene boje linka, dodatnog teksta, prikazanih simbola, itd) i
- sakrivanje linkova (eng. *link hiding*) – označava kao nedostupne ili nevidljive linkove za koje sistem na osnovu modela korisnika utvrdi da vode učenika ka nebitnim informacijama.

Sistem Protus poseduje elemente adaptivne hipermedije budući da prati putanju učenika kroz nastavni materijal tokom sesije i tako se utvrđuje status svakog pojedinačnog nastavnog resursa (dela lekcije) za trenutnog učenika. Sistem nudi dodatna objašnjenja uz pomoć uslovnih stranica – linkova koji se prikazuju, naglašavaju ili uklanjaju po potrebi, na osnovu informacija iz odgovarajućeg modela učenika. Iz modela učenika se utvrđuje da li pojedini resursi odgovaraju identifikovanom stilu učenja ili su predviđeni da predstavljaju učeniku neke potrebne informacije. Iz tog razloga, svim resursima u sistemu se dodeljuju odgovarajuće uloge koje nude informacije o njihovoj upotrebi. Na primer, ako resurs ima definisanu ulogu činjenice (*fact*) ili definicije (*definition*) tada se taj resurs koristi za povećanje osnovnog znanja učenika, a ako mu je dodeljena uloga primera (*example*) tada on služi da poveća učenikove praktične sposobnosti.

### **3.5 Modeliranje individualnih karakteristika učenika**

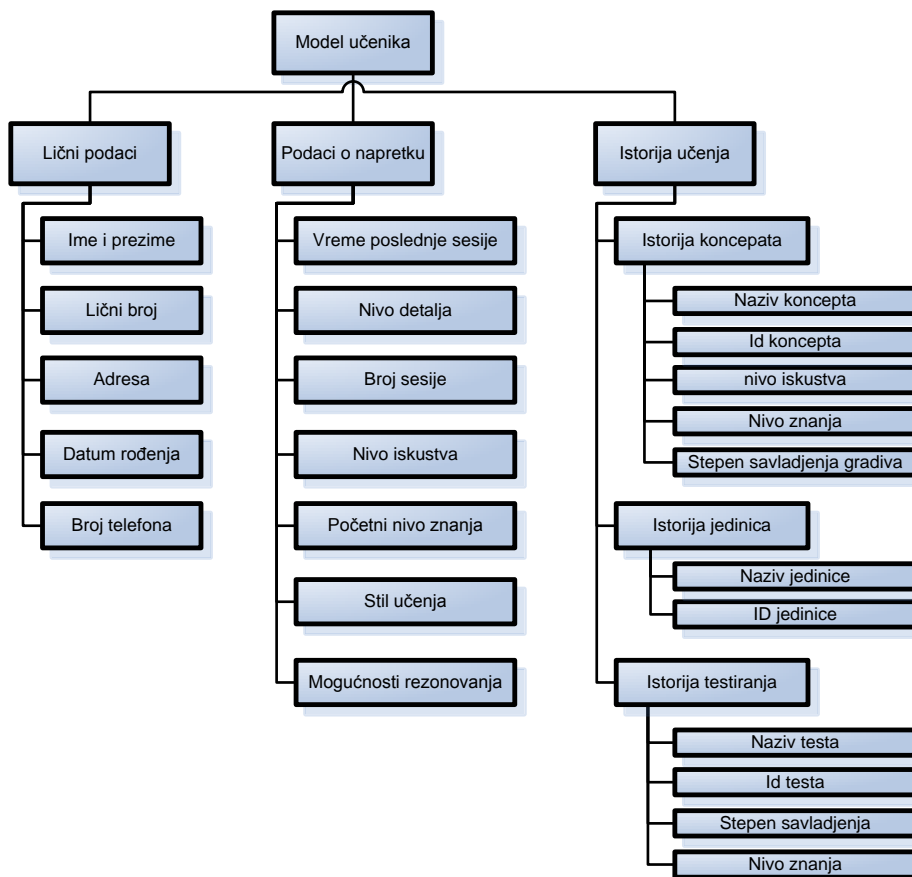
U svakom obrazovnom sistemu, model učenika predstavlja reprezentaciju objektivnih i subjektivnih informacija o učeniku, njegovim/njenim interakcijama, kao i prikaz napretka

učenika i istorija njegovih aktivnosti. Okruženje ili sistem upotrebljava te informacije u cilju maksimalnog povećanja mogućnosti predviđanja ponašanja učenika i adaptacije ka njegovim/njenim individualnim potrebama (Felder & Silverman, 1988; Parvez & Blank, 2008; Vesin et al., 2011b).

### 3.5.1 Izgled modela učenika u sistemima za elektronsko učenje

Model učenika predstavlja strukturu podataka prikazanu na slici 2. *Objektivne informacije* predstavljaju osnovne podatke o učeniku koje se koriste za njegovu identifikaciju (ime, prezime, adresa, datum rođenja, itd). Kao dodatak, objektivne informacije mogu uključivati i detalje vezane za početno znanje učenika, podatke o početnim ciljevima učenja i interesovanjima učenika, omiljenim medijima i kategorijama prezentovanja informacija učeniku, podatke o akcijama koje učenik pokreće prilikom pojedinih situacija i podatke o željenim reakcijama sistema na akcije učenika. Većina ovih informacija se prikuplja direktno od učenika, tako što se zatraži od njega/nje da popuni odgovarajuće upitnike ili obavi početno testiranje.

*Podaci o napretku učenika* u modelu učenika predstavljaju njegov/njen nivo znanja za odgovarajuću temu iz domena, učenikove greške, napredak i ukupan učinak na kursu. Većina ovih osobina se predstavlja kvantitativno i sistem ih automatski meri i ažurira. Beleženje i redovno ažuriranje ovih informacija u model učenika omogućuje sistemu da predvidi učenikove odgovore i reakcije na različite objekte učenja, predloži naredne aktivnosti, odredi zadatke, itd.



Slika 2. Grafička reprezentacija modela učenika

*Istorija učenja* uključuje informacije o kursevima i temama koje je učenik već obradio u sistemu, objektima učenja sa kojima je imao interakciju, testiranjima koje je prošao, itd. Kao i kod podataka o napretku učenika, sistem prati i beleži podatke o istoriji učenja.

Neki podaci u modelu učenika se ažuriraju redovno, prilikom svakog prijavljivanja na sistem, dok se neki ažuriraju retko ili samo jednom. Detalji modeliranja učenika će biti prikazani u narednim poglavljima.

Kvalitet modela učenika ne mora biti srazmeran količini podataka koji se u njemu čuvaju. Obično, što je veća količina podataka u modelu, veće su mogućnosti personalizacije i veća je preciznost modela. Ipak, povećanjem kompleksnosti modela, povećava se i problem njegovog ažuriranja.

## Poglavlje IV

### SEMANTIČKI WEB

Semantički veb (eng *Semantic web*) je nastao kao vizija Tima Berners-Lee-a<sup>1</sup>, globalne mreže (eng. *web*) kao univerzalnog medija za razmenu podataka, informacija i znanja (Berners-lee, 2000). On je 1999. godine izrazio svoju viziju semantičkog veba na sledeći način: „*Imam san o veb-u, u kojem računari postaju sposobni da analiziraju sve podatke na mreži – sadržaj, linkove i transakcije između računara i ljudi. Semantički veb, koji će sve to učiniti mogućim, tek treba da se razvije, ali kad se pojavi, svakodnevni mehanizmi trgovine, birokratije i naših svakodnevnih životnih aktivnosti biće obrađivani od strane računara. Inteligentni agenti o kojima se priča odavno, konačno će se materijalizovati*“.

Ljudi u svakodnevnim aktivnostima koriste veb kako bi uradili različite poslove kao što su: pronalaženje značenja stranih reči, rezervisanje knjiga u biblioteci, pretraga nižih cena pojedinih artikala, itd. Ipak, kompjuter ne može ispuniti te zadatke bez ljudskih uputstava budući da su veb stranice dizajnirane da ih čitaju ljudi a ne računari. Stoga je semantički veb pokušaj prikazivanja i memorisanja informacija koje su razumljive računarima tako da oni mogu odrađivati zahtevnije poslove uključene u pronalaženje, deljenje i razmenu informacija na mreži.

Semantički veb je nova generacija veba u kojem se pokušavaju predstaviti informacije na taj način da ih računari mogu koristiti ne samo za prikazivanje nego i za automatizaciju pretrage, integraciju i ponovnu upotrebu između aplikacija (Alsultanny, 2006). Cilj semantičkog veba je, dakle, da bude što razumljiviji računarima kako bi one pružile što kvalitetnije funkcionalnosti ljudima (Sheth et al. 2005). Takođe, jedan od bitnih ciljeva je i izgradnja odgovarajuće infrastrukture za inteligentne agente da izvršavaju kompleksne akcije po mreži. Da bi to radili, agenti moraju preuzimati i obrađivati odgovarajuće informacije, koje zahtevaju bezuslovnu integraciju agenata i mreže i potpuno iskorišćenje

---

<sup>1</sup> Tim Berners Lee je direktor W3C, World Wide Web Consortiuma. World Wide Web Consortium je važna međunarodna organizacija za standardizaciju na Internetu. Organizovana je kao konzorcijum gde organizacije članice učestvuju sa ljudstvom i resursima u cilju razvijanja standarda za World Wide Web.

prednosti koje nudi postojeća infrastruktura. Dalje, semantički veb predstavlja eksplicitno i inteligentno deklarirano znanje integrisano u veb aplikacije, uz semantički pristup Internetu i izdvajanje bitnih informacija iz teksta. Na kraju, semantički veb je način da se pouzdano implementira duboka integrisanost veb servisa, da se takvi servisi naprave razumljivim za računare, tj. da se kreira mreža od tesno povezanih servisa, razumljivih za računare koje će inteligentni agenti moći otkriti, izvršiti i automatski kombinovati (Berners-lee et al., 2001).

Problem sa današnjim vebom je što je on ogroman ali nedovoljno „pametan“ da lako integriše sve informacije koje poseduje a koje su potrebne korisniku. Takva integracija je potrebna u skoro svim oblicima upotrebe veba. Većina podataka sa veba je predstavljena u prirodnom jeziku i kao takva je narazumljiva za računare. Sa druge strane, ljudi mogu obrađivati samo mali deo informacija sa veba pa bi nam bilo korisno ako bi računari preuzeli deo posla oko obrade i analize sadržaja sa veba. Nažalost, veb je generalno orijentisan i namenjen ljudskoj upotrebi, tj. svi sadržaji na vebu su čitljivi za računare ali ne i razumljivi. Krajnjim korisnicima je potrebno da semantički veb prikazuje podatke na precizan, računarski razumljiv način, u formi pogodnoj da ih softverski agenti procesiraju, dele i ponovno koriste kao i da razumeju šta termini koji opisuju podatke znače. Eksplicitno predstavljanje semantike podataka bi omogućilo razvoj veba koji bi nudio potpuno nove oblike usluga kao što su na primer inteligentni načini pretrage ili efikasna razmena i filtriranje informacija.

Postoji niz značajnih koncepata koji omogućuju razvoj semantičkog veba. U narednim poglavljima će biti prikazani najvažniji od njih: ontologije, jezici semantičkog veba i pravila adaptacije.

#### 4.1 Ontologije

Reč *ontologija* je grčkog porekla i nastala je od reči *ontos* što znači biti, postojati i *logos* što znači nauka, učiti. U filozofiji ona predstavlja jedinicu postojanja tj. proučavanje osobina nečega. Tačnije, reč ontologija predstavlja proučavanje kategorija stvari koje postoje u nekoj oblasti ili domenu (Gruber, 1995).

Neformalno, ontologija određenog domena se tiče terminologije (rečnika domena), svih bitnih koncepata u domenu, njihove klasifikacije, taksonomije, njihovih veza (uključujući sve važne hijerarhije i ograničenja) i svih aksioma domena. Formalno, za nekog ko želi da raspravlja o temama iz domena D uz upotrebu jezika L, ontologija nudi katalog tipova elemenata koji postoje u domenu D. Ti tipovi u ontologiji su predstavljeni u obliku koncepata, odnosa (relacija) i predikata definisanih u jeziku L.

Bilo formalno ili neformalno, ontologija je veoma važan deo znanja bilo kojeg domena. Ontologija je često ključni deo znanja i svo drugo znanje se mora oslanjati i referencirati na nju.

Ontologije predstavljaju rečnik termina čija je semantika formalno definisana. One formalno i eksplicitno definišu koncepte koji su prisutni u okviru konkretnog domena, odnose između tih koncepata i njihove osobine (Gascuena et al., 2006). Takođe, ontologije su veoma korisne u različitim domenima budući da omogućuju ljudima i softverskim agentima razumevanje strukture prikazanih informacija i znanja. Omogućena je i ponovna upotreba ontologija i njenih pojedinih delova, tj. nije potreban razvoj nove ontologije ako je odgovarajuća ontologija već prisutna u tom domenu.



U veštačkoj inteligenciji termin ontologije predstavlja jednu od dve povezane stvari (Chandrasekaran et al., 1999):

- rečnik za prezentaciju, često specijalizovan za određeni domen ili temu,
- deo znanja koji opisuje neki domen uz pomoć nekog jezika za prezentaciju.

U oba slučaja, uvek postoji odgovarajuća struktura podataka koja predstavlja ontologiju.

Postavlja se pitanje kako bi ontologije trebalo da izgledaju. Odgovor na to pitanje zavisi od nivoa apstrakcije. Kada su implementirane u računaru, one obično izgledaju kao XML fajlovi. Budući da ontologije uvek predstavljaju neke koncepte i njihove međusobne odnose, mogu se grafički prikazivati uz pomoć nekog vizuelnog jezika. Grafički alati za izgradnju ontologija uvek podržavaju konverziju iz grafičkog formata u XML i druge tekstualne podatke.

Ljudi mogu prikazati ontologije kao skupove deklarativnih izjava u prirodnom jeziku. Ipak, izjave u prirodnom jeziku su teške za obradu uz pomoć računara. Ponovno prikazivanje na ekranu uz pomoć definicija koje predstavljaju ontologije takođe zahteva formalni jezik.

Tehnologije semantičkog veba pružaju osnovu razvoja nove generacije sistema za elektronsko učenje. Ontologije, definisane kao prikaz koncepata nekog domena, predstavljaju osnovnu komponentu semantičkog veba. Prve ideje o implementaciji ontologija u različitim aspektima sistema za elektronsko učenje prikazani su u (Mizoguchi & Bourdeau, 2000). Od tog momenta, veliki broj autora je predstavljao upotrebu ontologija u različitim aspektima učenja na daljinu, kao što su adaptivna hipermedija, personalizacija i modeliranje učenika (Jovanović et al., 2007; Henze, 2005; Merino & Kloos, 2008).

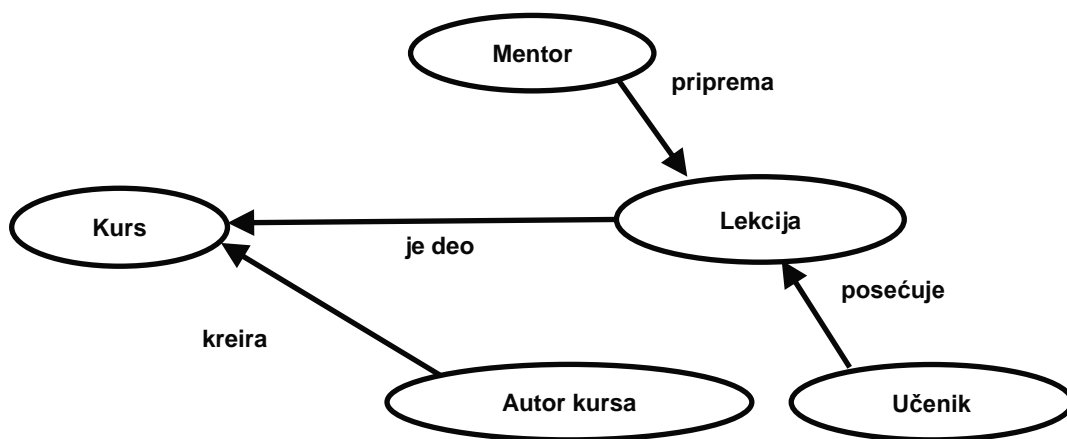
Potreba za razmenom i ponovnom upotrebom nastavnog materijala je uticala na rast popularnosti ontologija. Trenutni problem deljenja nastavnog materijala između sistema predstavlja činjenica da različiti sistemi koriste različite koncepte i termine za opis znanja iz domena (Swartout & Tate, 1999). Te razlike onemogućavaju preuzimanje nastavnog materijala iz već prisutnih sistema za elektronsko učenje. Javlja se potreba za definisanjem ontologija koje će predstavljati osnovu većeg broja sistema, čime bi se omogućila laka razmena informacija, ponovna upotreba nastavnog materijala i upotreba standardizovane terminologije. Razvoj takvih, ponovno upotrebljivih ontologija je važan cilj istraživanja iz oblasti semantičkog veba.

#### **4.1.1 Prikaz ontologija**

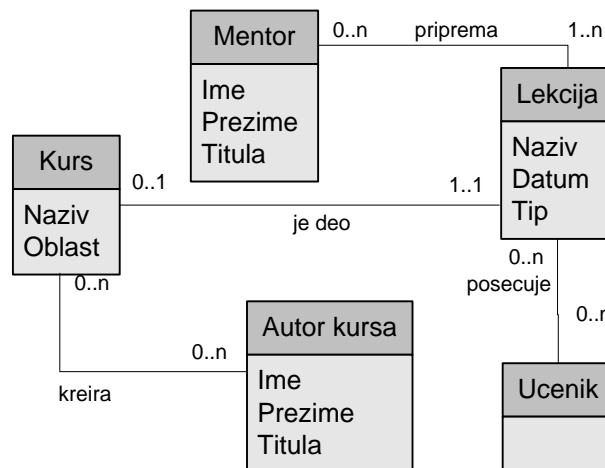
Ljudi prikazuju i razumeju ontologije kao niz deklarativnih izjava tj. rečenica. Takve rečenice su teške za procesiranje u računarima. Takođe, predstavljanje ontologija u računaru zahteva postojanje formalnih pravila. Budući da su ontologije, u stvari, predstavljanje koncepata i njihovih veza, one se mogu prikazati odgovarajućim dijagramom uz upotrebu odgovarajućeg vizuelnog jezika. Grafički alati za izgradnju ontologija poseduju funkcije konverzije grafičkog formata u XML ili druge tekstualne formate. Tako da ontologije predstavljene u računaru tipično izgledaju kao odgovarajući fajlovi bazirani na XML-u. To je posledica činjenice da i ostali jezici semantičkog veba koji se koriste u praksi i koji će biti prikazani u nastavku koriste XML sintaksu.

### Primer jednostavne ontologije sistema za elektronsko učenje

Neka je dat primer modelovanja kroz primer jednostavnog sistema za elektronsko učenje. Prikazan je samo deo sistema sa samo par najvažnijih veza između koncepata (slika 3). U tom sistemu postoji u upotrebi nekoliko osnovnih koncepata kao što su: učenik, kurs, lekcija, autor kursa i mentor. Neka postoje samo pojedine relacije između koncepata kao što su: autor kursa kreira kurs, mentor priprema lekcije, učenik posećuje lekcije koje su deo kursa. Ovo je niz deklarativnih rečenica koje predstavljaju koncepte ontologija i njihove relacije i koje su razumljive ljudima. Kao takve one nisu formalno definisane i neupotrebljive su za obradu računarima.



Slika 3. Ontologija sistema za elektronsko učenje prikazana kao semantička mreža



Slika 4. UML dijagram klasa sistema za elektronsko učenje

Na višem nivou apstrakcije, ova ontologija sa svojim konceptima se može neformalno prikazati dijagramom kao semantička mreža (slika 3).

Prethodni dijagram pati od mnogih nedostataka, kao što su neformalnost, nedostatak detalja, itd. Da bi se ova ontologija predstavila na formalniji način, može se koristiti UML dijagram (slika 4). Ovaj dijagram predstavlja isti deo sveta ali prikazan na formalniji način i sa više detalja.

Na slici 5 prikazana je ista ontologija ovog sistema za elektronsko učenje u ekvivalentnom formatu baziranom na XML-u uz pomoć jezika OWL (eng. *Web Ontology Language*) o kojem će biti više reči u nastavku poglavlja. Grafički editor ontologija sam automatski generiše odgovarajući deo koda. Baš ta reprezentacija ontologija se danas najčešće upotrebljava na nivou implementacije ontologija u sistemima.

```
<owl:Class rdf:ID="Kurs"/>
<owl:Class rdf:ID="Autor"/>
<owl:Class rdf:ID="Mentor"/>
<owl:Class rdf:ID="Ucenik"/>
<owl:Class rdf:ID="Lekcija"/>
<owl:ObjectProperty rdf:ID="kreira">
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="kreiranOd"/>
  </owl:inverseOf>
  <rdfs:domain rdf:resource="#Autor"/>
  <rdfs:range rdf:resource="#Kurs"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="priprema">
  <rdfs:range rdf:resource="#Lekcija"/>
  <rdfs:domain rdf:resource="#Mentor"/>
</owl:ObjectProperty>
...
<owl:DatatypeProperty rdf:ID="Prezime">
  <rdf:type
rdf:resource="localhost/ontologija/2012/0"/>owl:FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Ucenik"/>
...
</owl:DatatypeProperty>
...
```

Slika 5. Ontologija sistema za elektronsko učenje predstavljena u jeziku OWL

#### 4.1.2 Cilj upotrebe ontologija

Bez obzira što su ontologije dugi niz godina u upotrebi u raznim domenima i oblastima, često se pri odluci da se u nekom domenu koriste, postavlja pitanje: „Zašto ontologije?“ Jedan od odgovora bi mogao biti: zato što su one osnovni gradivni elementi sistema i aplikacija prilagođenih semantičkom vebu. Semantička saradnja između veb aplikacija moguća je samo ako je semantika veb podataka eksplicitno prikazana na vebu u formi računarski razumljivog domena i teorijskog sadržaja – ontologija. Uz automatsku upotrebu i računarsku interpretaciju ontologija, računari sami mogu ponuditi neograničenu podršku i automatizaciju pri pristupu i obradi informacija na vebu. Na taj način se skida teret sa korisnika i većinu posla obavlja sam semantički veb (Fensel et al., 2001).

Ontologije omogućuju pristup globalnoj kolekciji ljudskog znanja koje računari mogu razumeti i obrađivati. Kada se jednom znanje iz nekog domena postavi na veb u formi povezanih ontologija, ono kreira bazu za dalji razvoj inteligentnih aplikacija u tom konkretnom domenu jer olakšava problem prikupljanja znanja.

Ontologije imaju višestruke uloge u arhitekturi semantičkog veba (Horrocks et al., 2005):

- Ontologije omogućuju obradu, deljenje, i ponovnu upotrebu znanja sa veća uz pomoć aplikacija. To se postiže upotrebom zajedničkih koncepata i rečnika između različitih aplikacija.
- One omogućuju veći nivo interoperabilnosti na webu, tj. obezbeđuju različite oblike mapiranja termina unutar podataka, zato što ta mapiranja zahtevaju semantičku analizu.
- Omogućuju upotrebu inteligentnih servisa – agenata za pretragu, filtriranje informacija, inteligentnih integracija informacija, upravljanje znanjem, itd.

Uz pomoć ontologija se grade brojni korisni elementi inteligentnih sistema, kako pri opštem predstavljanju uopšte tako i pri procesu inženjeringa znanja. Neki od tih osnovnih elemenata su (Gruber, 1995):

- *Rečnik*. Ontologije nude rečnik za povezivanje termina iz odgovarajuće oblasti. Ovi rečnici se razlikuju od prirodnih rečnika (gde reči mogu različito biti tumačene u zavisnosti od ljudi koji ih tumače) jer nude logičke izjave koje opisuju šta određeni termini predstavljaju i u kakvoj relaciji mogu biti. Takođe, ontologije prikazuju pravila za kombinovanje tih termina i njihovih veza u cilju proširenja rečnika. Ontologije određuju termine na jedinstven i nedvosmislen način sa semantikom nezavisnom od okruženja ili osobe koja ih posmatra.
- *Hijerarhija koncepata*. Taksonomija ili hijerarhija koncepata predstavlja hijerarhijsku kategorizaciju ili klasifikaciju entiteta unutar odgovarajućeg domena. Svaka ontologija nudi taksonomiju na način razumljiv za računare. Relacije *je-nadklasa-od* su u ontologijama formalno definisane i uključuju formalne relacije među instancama i omogućuju konzistenciju pri upotrebi ontologija.
- *Teorija sadržaja*. Budući da ontologije identifikuju klase objekata, njihove veze i hijerarhije koje postoje u nekom domenu, one su, u stvari, autentične teorije sadržaja (eng. *content theory*). Ontologije ne samo da identifikuju te klase, veze i hijerarhiju, već ih i opisno određuju uz pomoć odgovarajućih jezika za izgradnju ontologija (eng. *ontology representation languages*). O ovim jezicima će kasnije biti više reči.
- *Provere konzistentnosti*. Dobro strukturirane i obrađene ontologije omogućuju različite provere ispravnosti i konzistentnosti (npr. provere tipova i vrednosti). Takođe omogućuju prenosivost između različitih aplikacija koje mogu te provere vršiti.
- *Razmena i ponovna upotreba znanja*. Najvažnija uloga ontologija nije prikaz rečnika i hijerarhije nego omogućavanje ponovne upotrebe znanja i njegovu razmenu između različitih aplikacija. Cilj je da svaka ontologija ponudi opis koncepata i veza koje postoje unutar domena i da oni budu na raspolaganju različitim inteligentnim agentima i aplikacijama.

Mada postoje mnogi jezici za prikaz ontologija i alati za razvoj baza znanja, ipak postizanje lake razmene i ponovne upotrebe ontologija nije jednostavno budući da postoje različite ontologije koje prikazuju isto znanje a pisane su u međusobno različitim jezicima.

#### 4.1.3 Oblasti primene ontologija

Sve brojne oblasti primene ontologija se mogu svrstati u četiri kategorije: saradnja, interoperabilnost, obrazovanje i modeliranje (Sheth et al., 2005).

*Saradnja.* Različiti ljudi mogu imati različite poglede na isti problem kada rade na nekom timskom projektu. Za njih, ontologije nude jedinstven kostur znanja kao smernicu daljeg razvoja. Još važnija uloga ontologija je uloga u saradnji između inteligentnih agenata u obliku njihove međusobne komunikacije. Razmena znanja između različitih agenata je jasnija kada su agenti svesni ontologija koje drugi agenti koriste kao modele.

*Interoperabilnost.* Ontologije omogućuju integraciju informacija iz različitih i odvojenih izvora. Krajnji korisnici obično ne mare na koji način će doći do informacija, važnije im je da dođu do svih informacija potrebnih za njih. Često su potrebne različite aplikacije da bi se pristupilo različitim izvorima znanja i da bi se stiglo do svih pristupačnih informacija. Ti različiti izvori mogu sadržati informacije u različitim formatima i na različitom nivou detalja. Stoga, ako svi izvori prepoznaju istu ontologiju, i konverzija podataka i integracija informacija se može lakše obavljati automatizovano.

*Obrazovanje.* Ontologije su takođe dobar medij za publikovanje naučnih radova i formiranje baze referenci. Budući da one obično nastaju na osnovu konsenzusa o znanju iz domena, ontologije mogu ponuditi pouzdane i objektivne informacije za one koji žele da nauče više o datom domenu.

*Modeliranje.* Pri modeliranju inteligentnih aplikacija, ontologije predstavljaju važne ponovno upotrebljive gradivne blokove koje mnoge aplikacije mogu da uključe kao već postojeće module znanja.

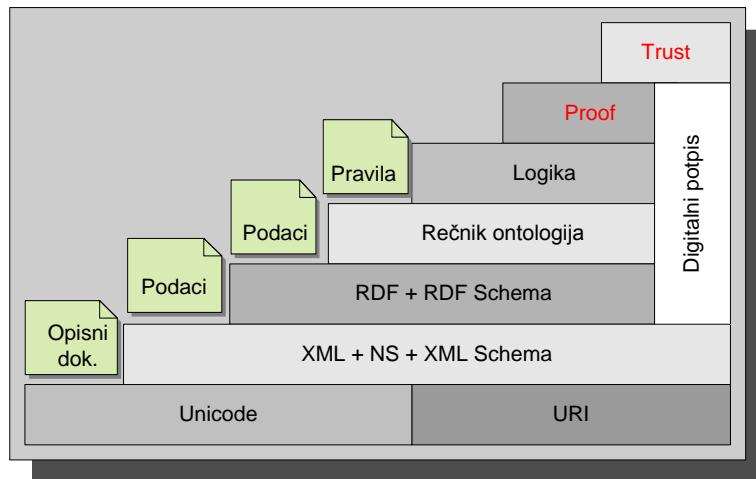
Mogle bi se odvojiti još neke kategorije upotreba ontologija kao što su upotreba u elektronskoj trgovini (omogućuju komunikaciju između prodavača i kupaca) kao i za automatizovanu pretragu.

## **4.2 Jezici semantičkog veba**

Tehnologije u sistemima za učenje na daljinu su česta tema pokušaja standardizovanja forme informacija u različitim komponentama tih sistema (Aroyo L. & Mizoguchi R, 2003). Ipak, razlike su i dalje prisutne budući da postoje različiti predlozi standarda i da se terminologije za opis resursa i aktivnosti razlikuju među njima.

U literaturi termini Jezici veb ontologija (eng. *Web-based ontology languages*) i Jezici semantičkog veba (eng. *Semantic Web languages*) se koriste kao sinonimi. Jezici semantičkog veba predstavljaju formalne jezike za kreiranje ontologija i samog semantičkog veba. Tehnologija semantičkog veba je sagrađena u slojevima, tj. vrši se u koracima (Dutta, 2006; Berners-Lee et al., 2001). Jezici višeg nivoa semantičkog veba koriste sintaksu i semantiku nižih nivoa (slika 6).

XML omogućuje korisnicima da dodaju proizvoljne strukture svojim dokumentima tako što će kreirati tagove radi obeležavanja veb stranica ili delova teksta. Iako je značenje XML tagova intuitivno jasno, oni sami ne omogućuju dodavanje semantike. XML se samo koristi kao prenosni mehanizam. *Resource description framework* - RDF i *Resource description framework schema* - RDFS nude osnovni okvir za prikazivanje metapodataka na vebu. Drugi jezici za prikazivanja znanja na vebu, kao što je OWL (bazirani na RDF-u), nude precizniju podršku reprezentaciji znanja.



Slika 6. Slojevi semantičkog veba

#### 4.2.1 XML - eXtensible Markup Language

Važan korak pre implementacije semantike i sintakse podataka u aplikacije je dogovor o njihovom jedinstvenom predstavljanju. Svi jezici semantičkog veba koriste XML sintaksu u cilju sprečavanja nepotrebnih i skupih modifikacija aplikacija radi premošćavanja razlika u semantici i sintaksi podataka. U suštini, XML je metajezik za reprezentaciju jezika semantičkog veba. Na primer, XML Schema definiše klasu XML dokumenata uz pomoć XML sintakse. RDF nudi okvir za predstavljanje meta podataka o veb resursima, i može biti prikazan u XML-u takođe. RDF Schema, OWL i drugi jezici veb ontologija takođe koriste XML sintaksu.

XML omogućuje specifikaciju i obeležavanje dokumenata razumljivih računarima. Slično kao i HTML, sastoji se od niza tagova koji služe da obeleže sadržaj dokumenta. Svaka XML Schema nudi neophodan okvir za kreiranje kategorija XML dokumenata.

XML ne nagoveštava specifičnu računarsku interpretaciju podataka. Informacije u XML dokumentima su samo predstavljene u jedinstvenoj sintaksi, ali njihova upotreba i semantika nije precizirana. Drugim rečima, XML predstavlja samo strukturu dokumenata, ne i njihovu računarsku interpretaciju. On nudi samo format struktuiranih dokumenata, bez preciziranja rečnika. Da bi se predstavilo znanje i semantika, moraju se koristiti precizniji jezici bazirani na XML-u.

#### 4.2.2 RDFS - Resource Description Framework Schema

RDFS ili RDF Schema je jezik za prezentaciju znanja koji nudi osnovne elemente za opis ontologija, takođe zvane RDF rečnici, u cilju struktuiranja resursa (RDF, 2012). Prvu verziju jezika je publikovao W3C u aprilu 1998 a konačne preporuke su objavljene u februaru 2004. Najvažnije RDFS komponente su uključene u izražajnijem jeziku OWL-u.

RDF je jezik koji omogućava predstavljanja podataka o pojedinim resursima na vebu u obliku trojki *objekat-atribut-vrednost* i semantičkih mreža, tehnike predstavljanja znanja iz veštačke inteligencije. Opis resursa u RDF-u predstavlja listu *izraza* (trojki), svaka prikazana u obliku *resursa* (objekat), njegovih *osobina* (atributa) i *vrednosti* tih osobina (vrednost). Vrednost može biti tekst ili drugi resurs. Svaki RDF opis može biti predstavljen kao usmereni obeleženi graf (semantička mreža), čiji delovi su ekvivalentni RDF izrazima.

RDF model nudi samo mehanizme nezavisne od domena za opis pojedinačnih resursa. On niti definiše semantiku domena aplikacije, niti donosi pretpostavke o posebnim domenima. Da bi se definisali elementi posebnih domena i njihova semantika, tj. ontologije, potrebna su druga okruženja. Sam RDF se koristi da se opišu instance ontologija, dok RDF Schema definiše ontologije.

RDF Schema (ili RDFS) nudi XML rečnik za definisanje klasa i njihovih veza, definisanje osobina i njihovo povezivanje sa klasama i omogućavanje kreiranja taksonomija. Da bi sve to omogućio, RDFS koristi modeliranje elemenata iz veštačke inteligencije, kao što su klasa (*class*), podklasa (*subClassOf*), osobine (*Property*) i podosobine (*subPropertyOf*).

RDF i RDFS omogućavaju opis činjenica o veb resursima, ali su često potrebni bogatiji i precizniji osnovni elementi da bi se definisala formalna semantika veb resursa. Na primer, u RDFS-u se ne mogu upoređivati klase i nemoguće je definisati ograničenja kardinaliteta.

#### 4.2.3 OWL - Ontology Web Language

OWL predstavlja familiju jezika za reprezentaciju znanja pogodnih za kreiranje ontologija, čiju je upotrebu odobrio *World Wide Web Consortium* (OWL, 2012). Ova familija jezika je bazirana na dvema (većim delom, ali ne u potpunosti kompatibilnim) semantikama: OWL DL i OWL Lite.

OWL rečnik uključuje skup XML elemenata i atributa, sa precizno definisanim značenjima. Oni se koriste za opis termina domena i njihovih relacija u ontologiji. OWL rečnik je baziran na RDF rečniku. Tako OWL sve elemente koje obrađuje deli na domen tipova (vrednosti koje pripadaju XML tipovima podataka) i domen objekata (pojedinačni objekti kao pojave klasa definisanih u OWL-u ili RDF-u). Takođe, postoje dve vrste OWL osobina:

- osobine koje povezuju objekte sa drugim objektima (definisanim sa *owl:ObjectProperty*)
- osobine koje povezuju objekte sa vrednostima tipova podataka (definisanim sa *owl:DatatypeProperty*)

OWL omogućava i definisanje dodatnih ograničenja i relacija između resursa, kao što su kardinaliteti, ograničenja domena, unija, presek, inverzna i tranzitivna pravila.

#### 4.2.4 SPARQL – Simple Protocol and RDF Query Language

Za razliku od jezika OWL i RDF, SPARQL nije namenjen za reprezentaciju ontologija i resursa, već za postavljanje upita u cilju selektovanja podataka na vebu. Tačnije, SPARQL je upitni jezik za RDF.

SPARQL se može koristiti za:

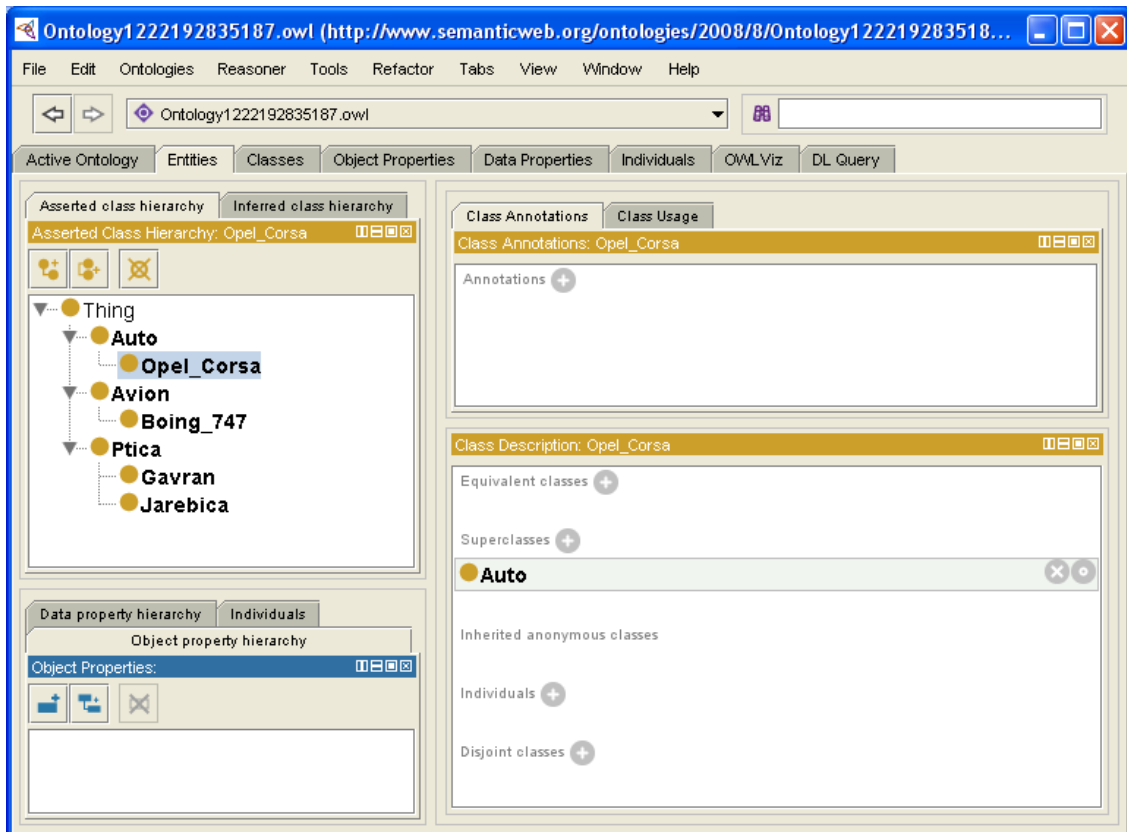
- selektovanje informacija iz RDF grafova,
- selektovanje RDF podgrafova i
- konstrukciju novih RDF grafova na osnovu informacija selektovanih iz postojećih RDF grafova.

Sintaksno, SPARQL upiti imaju sintaksu sličnu klasičnim upitnim jezicima baza podataka kao što je SQL. Stoga, ovi upiti poseduju i ključne reči SELECT i WHERE.

### 4.3 Grafička okruženja za razvoj ontologija

Grafička okruženja za razvoj ontologija integrišu editor ontologija sa drugim alatima i obično podržavaju upotrebu više jezika za prezentaciju ontologija. Cilj im je da ponude podršku za celokupan proces razvoja ontologija i njihovu kasniju upotrebu.

Trenutno, vodeći editor za razvoj ontologija je Protégé, razvijen na *Univerzitetu Stanford* (Protégé, 2012). Protégé je besplatan, open source editor ontologija i platforma za organizaciju znanja. Protégé omogućava definisanje koncepata ontologija (klasa), njihovih osobina, hijerarhija i različitih ograničenja kao i samih pojava klasa (slika 7).



Slika 7. Editor za razvoj ontologija Protégé

### 4.4 Obrazovne ontologije

Obrazovne ontologije su različite forme ontologija koje se koriste u sistemima za semantičko učenje na daljinu. Moguće je identifikovati nekoliko opštih kategorija obrazovnih ontologija (Aroyo & Mizoguchi, 2003; Chen, 2008; Mizoguchi et al., 2007):

- **Ontologije domena** (eng. *Domain ontologies*). Sistemi za semantičko učenje na daljinu ne mogu postići svoj cilj ako ne bi imali ontologije domena koje opisuju osnovne teorijske koncepte i relacije iz oblasti koja se podučava. Ontologije domena nisu važne samo za obrazovne namene već za sve aplikacije semantičkog veća. Budući da je cilj svakog sistema za semantičko učenje na daljinu napredak učenika u nekom domenu, autori kurseva i instruktori moraju znanje iz domena prikazivati uz pomoć ovog tipa ontologija
- **Ontologije akcija, zadataka** (eng. *Task ontology*). U svakom domenu aplikacija, a tako i u obrazovnom, ontologije akcija dopunjuju ontologije domena tako što



predstavljaju semantiku problema koji se rešavaju. Koncepti i relacije koji su uključeni u ontologije akcija predstavljaju vezu ka tipovima problema, strukturama, delovima, aktivnostima i koracima koje učenik mora da prati tokom procesa rešavanja problema. Na primer, ontologija akcija u obrazovnim aplikacijama može uključivati koncepte kao što su: problem, scenario, pitanje, odgovor, uputstvo, nagoveštaj, vežba, objašnjenje, simulacija, itd. Ontologije akcija u sistemima za semantičko učenje na daljinu formalizuju zadatke i aktivnosti svih bitnih učesnika u sistemu (učenika, predavača, autora), stoga su one ontologije dizajna instrukcija (eng. *instructional design ontologies*), ontologije kursa (eng. *training ontologies*), ontologije kreiranja zadataka (eng. *authoring task ontologies*), itd.

- **Ontologije nastavne strategije** (eng. *Teaching strategy ontology*). Ove ontologije nude instruktorima i autorima mogućnosti modeliranja nastavnih aktivnosti, tj definisanja znanja i principa koji se nalaze u pozadini pedagoških akcija. Na primer, ontologija nastavne strategije može definisati niz akcija koje se sprovede kada učenik napravi grešku, ili može definisati ponašanje sistema kojima se podstiče učenik da koristi alternativna rešenja.
- **Ontologije modela učenika** (eng. *Learner model ontology*). Dizajneri sistema za semantičko učenje na daljinu koriste koncepte iz ontologije modela učenika da bi izgradili model učenika. Ove ontologije i njima odgovarajući modeli učenika su najvažniji element adaptivnog ponašanja sistema. Sadržaj ontologije modela učenika najviše zavisi od aplikacije u kojoj se primenjuje. Tačnije, ontologija modela učenika treba da definiše koje objektivne i subjektivne podatke, kao i koje podatke o napretku učenika i istoriji učenja treba beležiti, čuvati i ažurirati za svakog pojedinačnog učenika.
- **Ontologije korisničkog interfejsa** (eng. *Interface ontology*). Svrha ovih ontologija je da definišu adaptivno ponašanje i tehnike sistema za semantičko učenje na daljinu na nivou korisničkog interfejsa. Stoga ontologije korisničkog interfejsa obezbeđuju eksplicitno modeliranje adaptivnosti za učenike sa različitim karakteristikama.
- **Komunikacione ontologije** (eng. *Communication ontology*). Različiti sistemi za semantičko učenje na daljinu, pedagoški agenti, obrazovni serveri i obrazovni servisi međusobno komuniciraju uz pomoć razmene poruka. Komunikacione ontologije definišu semantiku jezika sadržaja poruka, tj. rečnik termina koji se koriste u porukama.
- **Ontologije obrazovnih servisa** (eng. *Educational service ontology*). Ontologije obrazovnih servisa nude sredstva za kreiranje računarski čitljivih opisa servisa, posledice upotreba servisa i eksplicitnu reprezentaciju logike servisa. Obrazovni servisi imaju svoje osobine, sposobnosti, interfejs i rezultate, i oni se svi moraju zabeležiti u nedvosmislenoj, računarski razumljivoj formi da bi omogućili pedagoškim agentima da ih prepoznaju i automatski pokrenu.

Ontologije poseduju skup integrisanih mehanizama za postavljanje upita zasnovanih na adekvatnoj logici, ali je takođe neohodno uvođenje pravila adaptacije za generisanje korisnih principa i odnosa između pojedinih elemenata. Stoga, neophodno je uvođenje hijerarhije ontologija u cilju definisanja koncepata i relacija koji će se koristiti u pravilima, kao što je neophodno i definisanje sistema pravila čijom upotrebom će se selektovati

informacije iz ontologija. Pravila takođe nude izražajnost formalizma ontologija i postavljanja upita nad njima (Henze et al., 2004).

#### 4.5 Pravila adaptacije

Ontologije poseduju mehanizme za definisanja pravila i veza između entiteta (klasifikacije, relacije, identifikacije pojedinih entiteta, kardinaliteti). Međutim, često je neophodno definisati dodatna pravila, uslove i odnose između entiteta, koja ne mogu da se predstave samom definicijom ontologija (npr. ako je potrebno naglasiti da je savladavanje koncepta A neophodno pre prelaska na koncept B). Stoga je ontologijama neophodno dodeliti dodatni skup pravila koji će omogućiti definiciju i/ili pronalaženje novih informacija i pravilnosti u sistemu. Takođe, sistem pravila adaptacije zahteva prisustvo ontologija da bi se definisali koncepti i odnosi među njima koji se kasnije primenjuju u samim pravilima. Pravila takođe omogućuju postavljanje dodatnih uslova i odnosa među resursima kao i proveru da li su neki uslovi zadovoljeni (Henze et al., 2004).

Mehanizmi za obradu podataka integrisani u jezike semantičkog veba donekle ograničavaju upotrebu ontologija. Zato je prisutan pokušaj da se formalizuje logički sloj ontologija. *Semantic Web Rule Language* (SWRL, 2012) je predstavljen kao važan korak u ovom pravcu, čime se nadograđuje prethodni rad na razvoju *RuleML* jezika. Prisutnost standardizovanog jezika za definisanje pravila omogućiće istovremenu upotrebu ontologija i pravila adaptacije u cilju unapređenja aplikacija semantičkog veba.

##### 4.5.1 SWRL

SWRL je jezik posebno namenjen definisanju pravila adaptacije i zaključivanja u modelima znanja prikazanim preko OWL jezika (Sicilia et al., 2010). *Semantic Web Rule Language* (SWRL, 2012) je najpopularniji formalizam u veb zajednici za prikaz znanja u formi pravila. SWRL predstavlja kombinaciju jezika *Ontology Web Language* (OWL, 2012) i *Rule Markup Language* i uveden je od strane organizacije W3C kao kandidat za standard prikaza pravila u veb okruženju.

Pravila se definišu u formi implikacije između uslova (eng. *antecedent*) i posledice (eng. *consequent*). Značenje takvih pravila se može tumačiti kao: *ako su zadovoljeni uslovi definisani u telu pravila, tada uslovi definisani u posledici moraju takođe biti zadovoljeni*.

Osnovna prednost SWRL jezika je jednostavnost koju pruža kao proširenje izražajnosti OWL jezika. Druga prednost SWRL jezika je kompatibilnost sa sintaksom i semantikom OWL jezika, budući da se mogu kombinovati u istom logičkom jeziku.

Važno je napomenuti da većina postojećih aplikacija koje se baziraju na definiciji pravila primenjuju SWRL za definisanje pravila. SWRL nije previše izražajan (ne postoji negacija) niti je odlučiv, ali je najjednostavniji za upotrebu i primenu (Papataxiarhis et al., 2010).

Pravila adaptacije definisana u SWRL jeziku su oblika:

*antecedent*  $\Rightarrow$  *consequent*

i sastoje se od uslova definisanih u telu (eng. *antecedent*) i njihove posledice (eng. *consequent*) (SWRL, 2012). Neformalno, značenje pravila je: ako su uslovi definisani u telu pravila zadovoljeni, tada moraju biti zadovoljene i definisane posledice u istom pravilu adaptacije.

Detaljan opis formi sintaksnih pravila korišćenih prilikom izrade modela tutorskog sistema će biti prikazan u poglavlju 6.3 pod imenom Pravila adaptacije.

#### 4.5.2 Protégé

Protégé predstavlja *open source* editor za ontologije i okruženje za unos informacija u ontologije (Protégé, 2012). *Program* nudi intuitivni korisnički interfejs za jednostavni razvoj i ažuriranje ontologija kao i *SWRLTab* editor za definisanje SWRL pravila adaptacije. Uz pomoć *SWRLTab* komponente moguće je definisati SWRL pravila kao dodatak OWL ontologija.

Pored toga što je Protégé intuitivan i popularan alat, on takođe nudi *open-source Java API* koji omogućava pristup modelu ontologija i podacima iz ontologija iz Java programa.

#### 4.5.3 Jess

Obično se SWRL pravila transformišu u odgovarajući sistem za upravljanje pravilima (npr. Jess) koji omogućuje proveru istinitosnih stanja pojedinačnih pravila (Papataxiarhis et al., 2010).

Jess (Jess, 2012) je *Java framework* za editovanje i primenu pravila, budući da predstavlja okruženje za unos skriptova i manipulacije pravilima. U poslednje vreme, razvoj tehnologija za upravljanje pravilima je povećalo praktičnu vrednost sistema Jess među programerima veb aplikacija. Povrh svega, činjenica da je Jess sistem pisan u Javi, omogućuje njegovu integraciju sa brojnim već definisanim servletima i apletima. Jess podržava rad sa mnogim jezicima, kao što su RDF, OWL i SWRL. U cilju upravljanja pravilima, Jess koristi *SWRLTab* koji je jedan od plugin-ova za sistem Protégé (Lee et al. 2005).

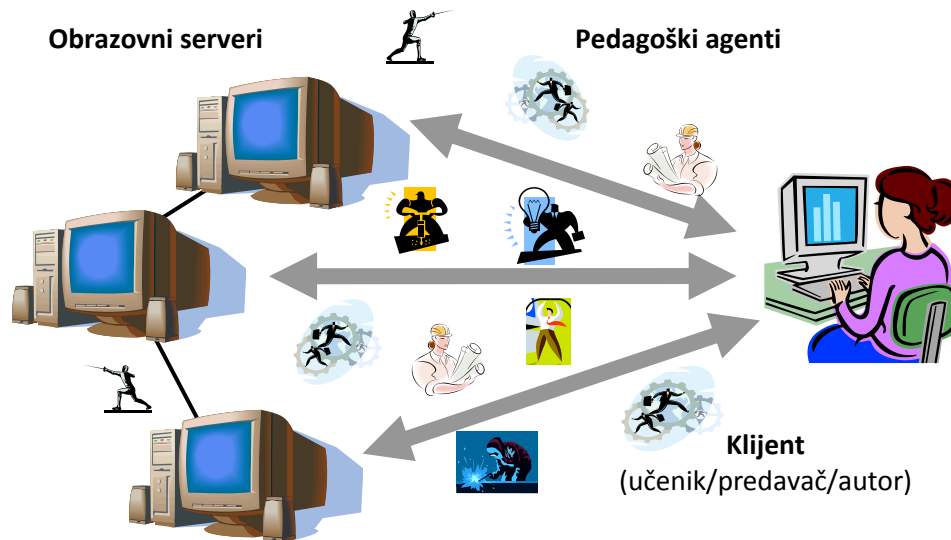
Definisana SWRL pravila se pokreću uz pomoć Jess's Java aplikacionog programskog interfejsa (eng. *application programming interface*) koji omogućuje kreiranje instance *Jess rule engine*-a. Instanca ove klase učitava sva definisana pravila i proverava uslove definisane u njima te pokreće izvršavanje pravila ako su svi uslovi ispunjeni. Nakon pokretanja odgovarajućih pravila, ažurirani podaci se zapisuju u OWL ontologije i na taj način se ažurira baza znanja.

### 4.6 Arhitektura semantičkih sistema za elektronsko učenje

Prvi neophodan preduslov razvoja semantičkih sistema za elektronsko učenje je adekvatan razvoj semantičkog veba. Potrebno je da semantički veb bude dovoljno rasprostranjen kao što je i sam veb trenutno. Naravno, to predstavlja dugotrajan i sistematičan proces. Paralelno sa procesom razvoja semantičkog veba, potrebno je i da sve više i više obrazovnog sadržaja bude pripremljeno za funkcionisanje u semantičkom okruženju. To podrazumeva razvoj velikog broja ontologija, opis obrazovnih sadržaja pomoću tih ontologija, razvoj posebnih obrazovnih servisa semantičkog veba, kao i razvoj različitih jezika semantičkog veba, alata i odgovarajućih tehnologija.

Na slici 8 je prikazano okruženje potrebno za memorisanje nastavnog materijala za podučavanje, testiranje, ocenjivanje i druge obrazovne aktivnosti u semantičkom vebu (Devedžić, 2004a; Shah, 2012; Mustapaşa et al., 2010). Ovo okruženje predstavlja generalizaciju arhitekture virtuelne učionice. Nastavni materijal je dostupan preko

različitih obrazovnih servera a posebne veb aplikacije odgovorne za upravljanje, administraciju i pristup materijalu se pokreću sa fizičkih servera. Učenici, predavači i autori kurseva pristupaju obrazovnom materijalu sa klijentske strane. Obrazovni sadržaj predstavlja proizvoljan materijal, pedagoški organizovan i strukturiran na takav način da ga zainteresovani učenici mogu koristiti da bi se upoznali sa znanjem iz domena, da bi ga razumeli i razvijali sposobnosti za rešavanje odgovarajućih problema. Inteligentni pedagoški agenti nude potrebne mogućnosti toka informacija, znanja i sadržaja između klijenata i servera.

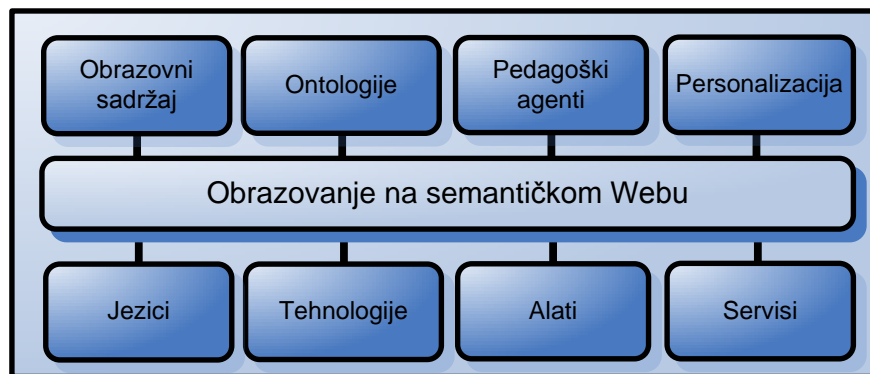


Slika 8. Okruženje za učenje u semantičkom vebu

Na slici 9 prikazani su osnovni koncepti sistema za učenje baziranog na semantičkom vebu. Slede njihove kratke specifikacije (Devedžić, 2006):

- **Autori** pripremaju **obrazovni sadržaj** u formi multimedijalnog nastavnog materijala, primera, upitnika, testova, vežbi, simulacija, itd. Ovaj sadržaj je obično strukturiran u obliku koherentnih jedinica za učenje, kao što su lekcije, poglavlja ili testovi.
- **Ontologije** predstavljaju osnovno znanje (kako znanje iz domena tako i pedagoško znanje) tako što definišu terminologiju, koncepte, relacije, hijerarhije koncepata i ograničenja. One omogućuju razmenu i ponovnu upotrebu obrazovnog sadržaja i saradnju različitih obrazovnih aplikacija.
- **Pedagoški agenti** pomažu pri lociranju, pretrazi, selektovanju, organizovanju, integraciji nastavnog materijala sa različitih obrazovnih servera. Oni se koriste i kao pomoć individualnom i grupnom učenju kao i podrška kognitivnim procesima učenika.
- **Učenici** su uvek zainteresovani za **personalizovano učenje** budući da svi ljudi imaju svoje lične navike, pristupe, ciljeve, želje i ritam učenja.
- **Različiti prirodni, vizuelni, i jezici za reprezentaciju** se koriste da dekodiraju i prikažu informacije koje se nalaze u materijalu za učenje. Takođe, različiti formalni jezici se mogu koristiti za razvoj obrazovnog sadržaja, za prikazivanje ontologija i obrazovnih servisa. Različiti jezici se koriste da definišu komunikaciju između različitih pedagoških agenata.

- Iako **tehnologija** nije krajnji cilj sistema za elektronsko učenje, njena upotreba je ključni faktor razvoja. Trendovi i realnost trenutne tehnološke podrške ne smeju se ignorisati prilikom izrade sistema za elektronsko učenje.
- **Razni alati za učenje, predavanja i kreiranje kurseva** često dolaze kao moćne integrisane veb softverske aplikacije. Takođe, postoje brojni drugi softverski alati različitih mogućnosti za pripremu obrazovnog sadržaja.
- **Servisi semantičkog veba** se koriste da ponude predavačima, učenicima i autorima kurseva pristup obrazovnom sadržaju iz različitih domena interesovanja. Oni su obično povezani sa obrazovnim serverima i mogu podržavati brojne različite obrazovne aktivnosti.



Slika 9. Koncepti sistema za učenje uz korišćenje semantičkog veba

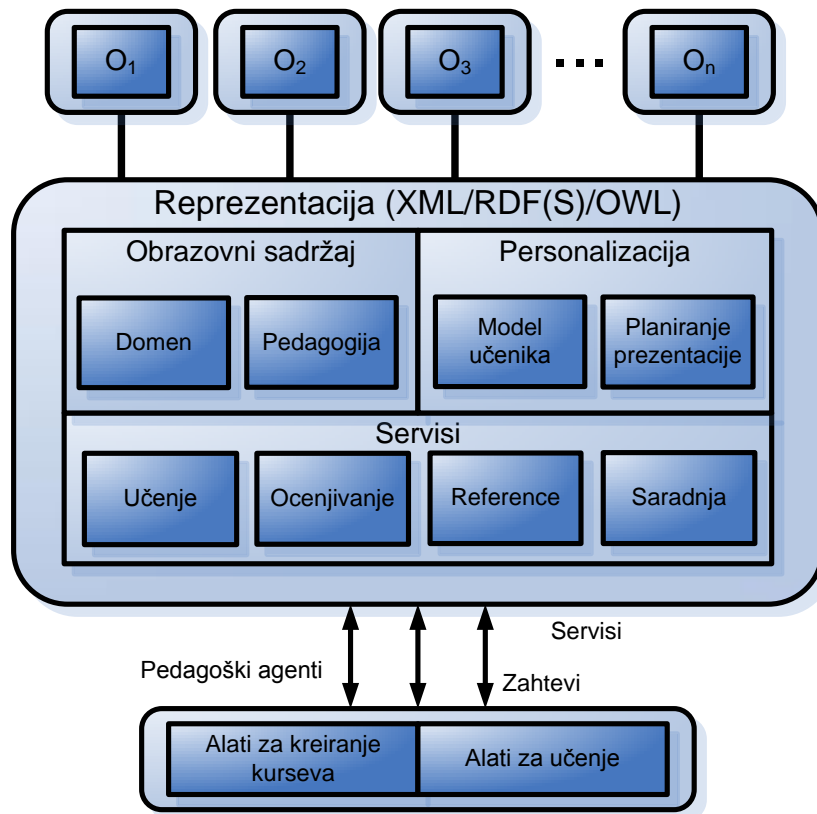
Na slici 10, prikazan je opšti model obrazovnih servera za sisteme za semantičko učenje na daljinu (Shah, 2012; Devedžić, 2004b). Obrazovni server bi trebalo da koristi inteligentne tehnike u cilju personalizacije nastavnog materijala. Server bi trebalo da poseduje planer personalizacije koji pomaže da inteligentni tutor odabere, pripremi i prilagodi materijal iz domena koji će prikazati učeniku u skladu sa njegovim stilom i načinom učenja.

Tutor postepeno gradi model učenika tokom sesija u cilju praćenja učenikovih aktivnosti i procesa učenja, pronalaska i ispravljanja grešaka koje je učenik napravio i po mogućnosti preusmeri sesiju na odgovarajući način. Na kraju sesije, obavlja se snimanje podataka u model učenika. Nakon toga se tako prikupljeni podaci koriste sa već postojećim podacima o učeniku prilikom formiranja naredne sesije sa istim učenikom.

Iz perspektive predavača, obrazovni server omogućuje pristup, pregled i pretragu kolekcije objekata učenja i pristup resursima vezanim za selekciju i prilagođavanje nastavnog materijala. Server takođe nudi mogućnosti praćenja i izmene rada različitih klasa učenika kao i direktan pristup modelima učenika, kako u toku tako i nakon izvođenja sesija. Server sâm može nuditi alate za obavljanje osnovnih aktivnosti predavača ili može koristiti klijentske alate za konekciju sa obrazovnim serverom i upotrebljavati servise koje taj server nudi.

Iz ugla autora kurseva, obrazovni serveri proširuju alate namenjene autorima. Uz pomoć alata za autore, autori domena pristupaju različitim ontologijama koje se čuvaju u bibliotekama ontologija na obrazovnom serveru. Autori mogu definisati svoje ontologije i publikovati ih u biblioteke. Oni, takođe mogu ponovno upotrebljavati postojeće ontologije iz biblioteka i vršiti na njima potrebne izmene. Autori mogu i da kreiraju,

sačuvaju, ažuriraju i brišu objekte učenja, kurseve, menjaju instrukcioni dizajn, pristupaju modelu učenika i adaptiraju ga ili proširuju, itd.



Slika 10. Model obrazovnog servera

Potrebno je obezbediti individualni prolazak kroz obrazovni materijal samo uz interakciju sa sistemom, kao i mogućnosti saradnje sa drugim učenicima. Bitni elementi su i lakoća pristupa sistemu i samom materijalu za učenje i filtriranje sadržaja za pojedinačnog učenika. Ontologije tu služe da se obezbedi personalizacija i semantička pretraga i integracija sadržaja iz različitih izvora. Često korisnici nisu ni svesni do koje granice se šire ontologije i odakle ih preuzimaju. Njima je dovoljno da dobiju odgovarajući materijal u odgovarajuće vreme i da im je prezentovan na intuitivnan i razumljiv način.

U cilju personalizacije procesa učenja i prilagođavanja sadržaja svakom pojedinačnom učeniku, sistemi za semantičko učenje na daljinu moraju koristiti strategije kojima će zadovoljiti potrebe pojedinačnog učenika. Takođe, ti sistemi moraju koristiti različite tehnologije radi promene okruženja i adaptacije nastavnog materijala na osnovu potreba učenika (Devedžić, 2006). Proces adaptacije može biti u formi adaptacije sadržaja, procesa učenja, povratnih informacija ili navigacije. Kao rezultat takve adaptacije se može desiti da jedna lekcija može biti prezentovana različitim učenicima uz pomoć dva potpuno različita skupa nastavnog materijala.

Uzmimo, na primer, slučaj sistema za elektronsko učenje programskog jezika Java, gde učenik prolazi kroz niz lekcija i obrađuje delove teorije o osnovama sintakse programskog jezika Java i pri tom testira svoje novostečeno znanje (Vesin et al., 2012a). Ako bi jedan učenik želeo da nauči deo sintakse o upotrebi naredbe za grananje, sistem bi prvo prikazao nastavni materijal kojim bi objasnio pravila i sintaksu upotrebe te naredbe i zatim bi mu prikazao niz adekvatnih, ilustrativnih rešenih primera. Ako bi neki drugi

učenik želeo da nauči praktičnu primenu te naredbe, sistem bi mogao eventualno da testira znanje učenika iz osnovnih koncepata sintakse programskog jezika Java (npr. definisanje izraza ili deklaracije promenljivih) i tek potom mu predstavi detalje o sintaksi i upotrebi naredbi grananja. Na kraju procesa sistem bi mogao eventualno da testira znanje koje je učenik usvojio iz te lekcije.

## **Poglavlje V**

### **POREĐENJE SISTEMA PROTUS 2.0 I SLIČNIH TUTORSKIH SISTEMA**

Savremene informacione tehnologije nalaze značajnu ulogu i intenzivno se koriste prilikom izrade različitih tipova obrazovnog softvera, počev od prvobitnih elektronskih tutorijala, preko multimedijalnih onlajn kurseva sve do obrazovnih kompjuterskih igara. U ovoj distertaciji, pažnja je usmerena samo na nekoliko specifičnih tipova tutorskih sistema:

- tutorski sistemi koji prilagođavaju nastavni materijal specifičnim stilovima učenja učenika,
- tutorski sistemi koji koriste različite tehnike za generisanje preporuka sa ciljem da preporuču učeniku odgovarajuće onlajn aktivnosti na osnovu njihovih potreba, želja, znanja kao i putanja kretanja kroz nastavni materijal koji su koristili drugi učenici sa sličnim karakteristikama i stilom učenja,
- tutorski sistemi koji su bazirani na tehnologijama semantičkog vebe.

#### **5.1 Sistemi sa adaptacijom na individualne stilove učenja učenika**

Različiti učenici na različit način pristupaju i koriste nastavni materijal, uče i usvajaju novo gradivo. Stoga je poželjno da sistemi za elektronsko učenje ponude isti nastavni materijal na različite načine, prilagođene svakoj od kategorija potencijalnih učenika sistema. Postoje mnogi modeli stilova učenja u raspoloživoj literaturi od kojih svaki donosi različite opise i klasifikacije stilova učenja. Neki od najpoznatijih su modeli Coffield-a (Coffield et al 2004), model Kolb-a (Kolb, 1984) i Dunn and Dunn model (Dunn et al., 1984).

Sistemi koji prilagođavaju nastavni materijal stilovima učenja koriste različite metode za utvrđivanje kojoj kategoriji učenici pripadaju. Jedan od načina je da učenik popuni upitnik koji u različitim formama postoji za svaki model stilova učenja (Parvez & Blank, 2008; Peňa et al., 2002; Wang, Li & Chang, 2004; Wolf, 2003). Drugi metod je određivanje stila učenja na osnovu interakcije učenika sa tutorskim sistemom. Sistem prati tok učenja, posećene stranice i linkove koje pokreće. Na osnovu tih rezultata, sistem formira bazu



podataka posećenog nastavnog materijala i donosi zaključke koji materijal odgovara učenicima sa odgovarajućim stilom učenja. Takav pristup je primenjen u sistemu prikazanom u (Kelly & Tangney, 2004). Nakon što se za konkretnog učenika utvrdi stil učenja, ovi sistemi prikazuju nastavni materijal u formi koja je prilagođena za taj stil učenja ili u odgovarajućem rasporedu prilagođenom za taj stil učenja.

Adaptivnost postojećih sistema je često realizovana u obliku različitog prikaza slika i teksta sa izuzetkom sistema iWeaver (Wolf, 2007) koji nudi i odgovarajući multimedijalni materijal. Takođe, sistemi uglavnom prilagođavaju nastavni materijal stilu učenika koji je određen na početku korišćenja sistema. U nastavku kursa se pretpostavlja da će za konkretnog učenika njegov stil ostati nepromenjen. Ipak, istraživanja pokazuju da se stilovi učenja pojedinačnih učenika menjaju u zavisnosti od trenutnih ciljeva ili kvaliteta konkretnog nastavnog materijala. Tako da se čini kontraproduktivno da se u toku kursa učenik veže samo za jednu kategoriju materijala, tj. da se tokom sesija dodatno ne preispituje odluka o dodeljenom stilu učenika.

Kada se radi o sakupljanju i obradi informacija o individualnim stilovima učenika u tutorskim sistemima mnoga pitanja su i dalje otvorena. To su uglavnom pitanja koja se tiču načina na koji učenici sa različitim stilovima učenja odgovaraju na pitanja postavljena na testovima, izvode vežbe i razne druge nastavne aktivnosti. Takođe, pitanje je kako stilovi učenja utiču na kretanje kroz nastavni materijal, koje su zajedničke osobine učenika sa pojedinačnim stilovima učenja, na koji način pojedine kategorije učenika biraju i koriste nastavni materijal, itd (Vesin et al. 2013).

## **5.2 Sistemi sa ugrađenim podsistemima za generisanje preporuka**

Sistemi za generisanje preporuka (eng. *recommendation systems*) koriste mišljenja različitih korisnika sistema u cilju identifikacije potencijalnih interesovanja pojedinačnih korisnika (Resnick et al., 1994). Neki od takvih sistema su primenjeni u sistemima za elektronsko učenje za generisanje preporuka koji nastavni materijal treba prikazati učeniku u cilju što bržeg savladavanja gradiva (Kristofic, 2005; Farzan & Brusilovsky, 2006).

Istraživanje o upotrebi sistema Netlogo za generisanje preporuka učenicima je predstavljeno u (Koper, 2005). Autor je predstavio različite forme prikaza nastavnog materijala koje utiču na povećanje ili smanjenje motivacije kod učenika prilikom korišćenja sistema za elektronsko učenje. Slična komponenta za generisanje preporuka je ugrađen u sistem predstavljen u (Janssen et al., 2007). Međutim, preporuke generisane u ovom sistemu ne uzimaju u obzir lične karakteristike učenika.

Metodologija poboljšanja već gotovih onlajn kurseva uz pomoć adaptacije je predstavljena u (Romero et al., 2004). Takozvani AHA! projekat je prvobitno zamišljen kao podrška onlajn kursevima prilikom implementacije adaptivnosti hipermedijalnih kurseva na Tehnološkom univerzitetu u Ajndhovenu (*Eindhoven University of Technology*) (Romero et al., 2007). Najvažniji doprinos ovog projekta je adaptacija prezentacionog i navigacionog sistema kursa na osnovu nivoa znanja učenika. AHA! arhitektura primenjuje posebna predikciona pravila za postizanje adaptacije.

Istraživanje o implementaciji tehnika *data mining*-a u sisteme za elektronsko učenje i poboljšanjima procesa učenja koji se time postiže je predstavljen u (Tang & McCalla, 2005). Autori su predložili tehniku *data clustering*-a kao način modelovanja učenika.

Metoda clustering-a je prikazana i u (Hammouda & Kamel, 2006) u cilju grupisanja nastavnog materijala na osnovu teme koje obrađuju i sličnosti u njihovoj formi. Tehnike Data mining-a kao što su *Association Rule mining* i *Intra-session frequent pattern mining* su primenjene u (Zaïane, 2002) u cilju izdvajanja korisnih modela ponašanja učenika. Ti modeli su korisni administratorima sistema i nastavnicima u cilju evaluacije aktivnosti učenika tokom onlajn kursa.

Svi ovi sistemi su prikazani kao inovativni i napredni ali na žalost ni jedan od tih tutorskih sistema nije ušao u širu upotrebu. Svrha ovih sistema je prikaz nastavnog materijala iz različitih domena čime je otežana implementacija nekih od specifičnih funkcionalnosti koje zahtevaju sistemi za učenje programskih jezika, kao što su: onlajn programiranje, prikaz sintaksnih pravila, itd.

Postoji niz tutorskih sistema namenjenih učenju programiranja u programskom jeziku Java ali većina ne koristi sve mogućnosti adaptacije nastavnog materijala (Emurian, 2006; Holland et al., 2009; Sykes & Franek, 2003), najviše zato što očekivana komercijalna zarada takvih sistema ne opravdava troškove izrade sistema i adaptivnih kurseva. Ni jedan od prethodno navedenih sistema ne koristi sve mogućnosti tehnika generisanja preporuka kao što su *collaborative filtering*, *association rule mining* i klasterovanje nego samo najosnovnije tehnike *data mining*-a. Takođe, retki su i sistemi koji uzimaju u obzir razlike koje postoje među učenicima sistema kao i postojanje različitih stilova učenja kod pojedinih učenika.

Sistem Protus se razlikuje od sistema prikazanih u prethodno navedenim radovima u nekoliko osnovnih aspekata:

- Protus koristi nekoliko tehnika personalizacije koje omogućavaju prilagođavanje nastavnog materijala (upotreba sistema za generisanje preporuka, identifikacija stilova učenja, itd.).
- Pored sistema za generisanje preporuka takođe se koriste i personalizacija na osnovu identifikacije stilova učenja.
- Sistem podržava klasterovanje nastavnog materijala i učenika i opcije generisanja optimalne putanje učenika kroz nastavni materijal.
- Protus nudi mogućnosti unosa lekcija iz drugih oblasti (ne samo programiranja) čime se omogućava kreiranje više kurseva iz različitih oblasti.
- Arhitektura sistema u formi ontologija omogućuje opštost arhitekture i mogućnost unosa nastavnog materijala u različitim formama (tekstualni, grafički, itd.). Na taj način omogućen je prikaz informacija i lekcija nezavistan od odabranog domena.

### **5.3 Tutorski sistemi sa elementima tehnologija semantičkog veba**

Mnoga istraživanja su sprovedena sa temom upotrebe tehnologija semantičkog veba prilikom izrade sistema za elektronsko učenje ali većina razvijenih sistema koristi ontologije samo za prikaz nastavnog materijala i podataka o učenicima (Jia et al., 2011; Fernandes-Breis et al., 2012; Gascuena et al., 2006; Hee Lee et al., 2002).

Prototip sistema SMARTIES se u potpunosti zasniva na tehnologijama semantičkog veba (Mizoguchi et al., 2007). Najnovija verzija sistema sadrži ontologije koje prikazuju samo apstraktni dizajn nastavnog materijala, tj. nije još konkretizovan sadržaj lekcija i

celokupno znanje iz domena. Protus 2.0 za razliku od ovog sistema ne samo da primenjuje modeliranje sadržaja nego i povezuje semantiku i sadržaj lekcija uz pomoć implementacije niza pravila adaptacije.

Sistem *Personal Reader* predstavlja još jedan važan rezultat upotrebe tehnologija semantičkog vebe u cilju personalizacije nastavnog materijala (Dolog & Nejd, 2007; Henze et al., 2004). Autori predlažu arhitekturu baziranu na RDF-u (eng. *Resource Description Framework*) i ontologijama koje prikazuju nastavne resurse, znanje iz domena kao i podatke o samim učenicima. U (Hence et al., 2004) autori predstavljaju nekoliko ontologija za izgradnju adaptivnog obrazovnog hipermedijalnog sistema. Međutim, strategija adaptacije primenjena u sistemu nije predstavljena ontologijama. Ontologija nastavne strategije gradiva je prikazana u sistemu Protus 2.0 da bi se jasno izdvojile aktivnosti koje su deo generisanja preporuka učenicima.

Po funkcionalnostima i nameni, sistemu Protus 2.0 je najbliži sistem QBLS, predstavljen u (Dehors & Faron-Zucker, 2006). QBLS je inteligentni tutorski sistem prilagođen radu na mreži u kome su primenjene tehnologije i standardi semantičkog vebe. Sistem preuzima nastavne resurse pronađene na Internetu i služi kao podrška učenju programskog jezika Java. Za razliku od ovog sistema koji sakuplja nestruktuirani nastavni materijal sa Interneta, Protus 2.0 implementira celokupan Java programerski kurs. Iz ugla arhitekture, prednost sistema Protus 2.0 je što su svi elementi njegove arhitekture predstavljeni u formi ontologija dok u sistemu QBLS ontologije sadrže samo nastavni materijal.

Autori u (Fernandes-Breis et al, 2012) su prikazali sistem koji su razvili nastavnici srednje škole i koji koristi ontologije za razvoj i ažuriranje nastavnih planova za svoje časove. Oni opisuju softversku platformu zvanu Gescur koja implementira procese formiranja, planiranja, usmeravanja i kontrole nastavnih planova i programa.

Experiment sa integracijom modela znanja iz domena dva različita adaptivna sistema je prikazan u (Sosnovsky et al., 2008). U radu su takođe prikazani organizacija i struktura ontologije domena. Ekperimentalna ontologija Java programerskog kursa (*Java Learning Object Ontology – JLOO*) je data u (Lee et al., 2005). Definisana ontologija je korišćena u radu kao osnova za razvoj nastavnog materijala početnog kursa iz Java programskog jezika kao i za integraciju tog materijala u adaptivni sistem za elektronsko učenje.

Dodatni predlozi upotreba ontologija prilikom izgradnje pojedinih elemenata sistema za elektronsko učenje, kao što su model učenika, ontologija domena, ontologija akcija i dr. se mogu naći u (Aroyo & Mizoguchi, 2003; Lee et al., 2005; Sosnovsky et al., 2008). Pored toga, ovi sistemi ne definišu ostale komponente arhitekture tutorskih sistema niti komunikaciju između njih. U sistemu Protus 2.0, sve komponente arhitekture su predstavljene tehnologijama semantičkog vebe.

U strukturi prethodno prikazanih sistema, upotreba ontologija se uglavnom fokusira na modeliranje znanja iz domena i formiranje nastavnog materijala dok su definicija i komunikacija između drugih komponenti sistema izostavljene. Jedan od osnovnih ciljeva ovog rada je i prikaz tutorskog sistema u formi više ontologija kao unapređenja nedostataka postojećih sistema.

### 5.3.1 Upotreba pravila adaptacije u tutorskim sistemima

Nekoliko radova novijeg datuma prikazuje način na koji se pravila adaptacije mogu koristiti u kombinaciji sa ontologijama u kreiranju i realizaciji procesa adaptacije u sistemima za učenje (Sicilia et al., 2010; Chi, 2009; Popescu et al., 2007). Autori u (Sicilia et al., 2010) prikazuju reprezentaciju osnovnih elemenata instrukcionih modela uz pomoć formalnog jezika ontologija OWL u kombinaciji sa SWRL jezikom. U radu je prikazana upotreba ontologija koje opisuju formalno sekvencu i strukturu aktivnosti učenika u sistemu i ulogu svakog resursa nastavnog materijala pri tim aktivnostima.

Istraživanje prikazano u (Chi, 2009) definiše raspored nastavnog materijala u sistemu za pohađanje početnog matematičkog kursa. Autori predlažu dve osnovne funkcionalnosti:

- prikaz apstraktnih pogleda na definisanje rasporeda nastavnog resursa i nastavnog materijala uz pomoć ontologija i
- semantička pravila za prikaz odnosa između osnovnih elemenata nastavnog materijala.

Detaljan opis pravila za identifikovanje stilova učenja i pravila adaptacije na osnovu identifikovanih stilova je prikazan u (Popescu et al., 2007). Autori su prikazali pravila u neformalnoj pseudokod notaciji da bi sistem predstavili u formi nezavisnoj od formalizma.

Ovi radovi pokazuju da je najveća poteškoća modelovanje znanja u sistemima za elektronsko učenje, kreiranje i ažuriranje struktura semantičkog veba (npr. ontologija). Te strukture mogu biti korišćene ne samo za organizovanje nastavnog materijala nego i za definisanje odnosa među njima i definisanja redosleda prikaza materijala učeniku. U sistemu Protus ontologijama nije predstavljeno samo znanje iz domena nego i aktivnosti tokom izvođenja personalizacije i adaptacije u sistemu.

## 5.4 Primeri tutorskih sistema

U svim prethodno navedenim sistemima, elementi arhitekture sistema su implementirani tehnologijama semantičkog veba ali su detalji primenjene personalizacije izostavljeni. Iz tog razloga, u sistemu Protus 2.0 akcije personalizacije su prikazane kroz implementaciju odgovarajućih ontologija i definisanjem pravila adaptacije. Nekoliko karakterističnih primera sistema koji koriste tehnologije semantičkog veba za generisanje personalizovanog kursa su prikazani u nastavku.

### 5.4.1 iWeaver

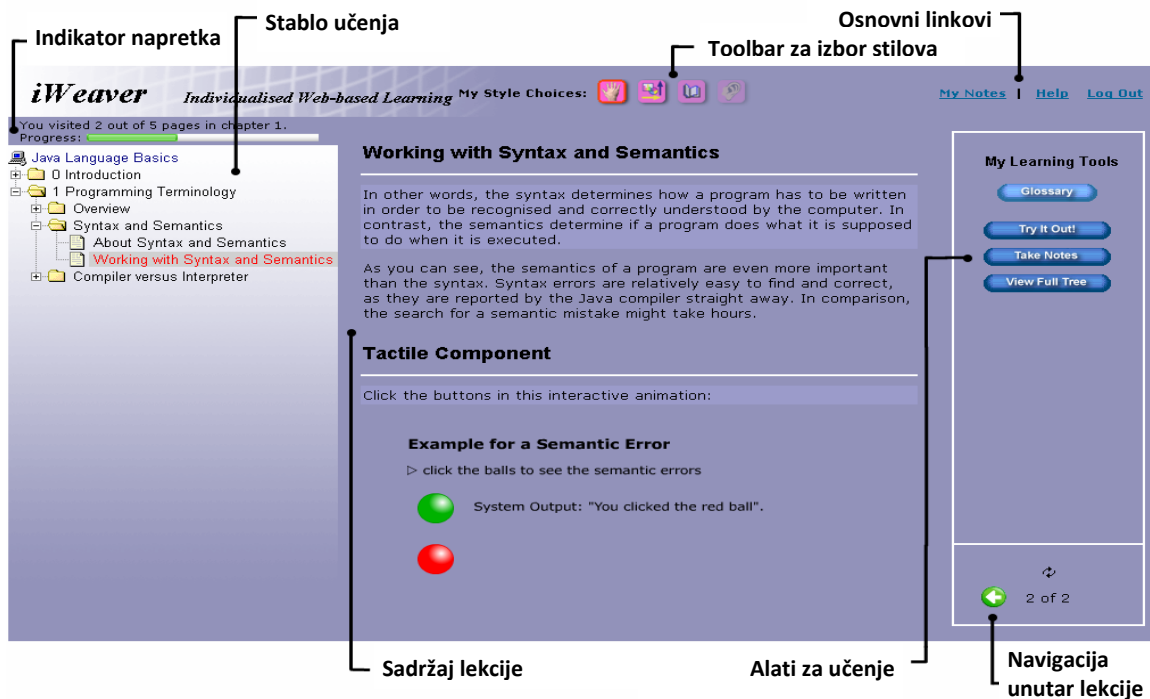
*iWeaver* predstavlja sistem za primenu individualizacije pri učenju programskog jezika Java. Sistem je nastao kao rezultat projekta za realizaciju personalizovanog sistema i izrade doktorske disertacije čiji je autor Christian Wolg sa Fakulteta za obrazovanje, jezike i društvene nauke u Melburnu, Australija (*Faculty of Education, Language and Community Studies, Melbourne*) (Wolf, 2007). Sistem preko JSP (*Java Server Pages*) stranica prikazuje različite mutimedijalne elemente i druge formate za prikazivanje lekcija iz programiranja. U *iWeaver*-u je implementirana nekolicina adaptivnih tehnika kao što su: sortiranje linkova, sakrivanje linkova i uslovni sadržaji stranica (Wolf, 2003).

**Upotreba sistema** - Kada učenik pristupi *iWeaver* sistemu po prvi put, od njega se zahteva da popuni standardizovani upitnik (eng. *Building Excellence Inventory*). Na

osnovu upitnika koji sadrži 118 pitanja za ponuđenim odgovorima na osnovu *Dunn&Dunn* modela stilova učenja (Dunn et al., 1984), procenjuje se početni stil učenja i kreira njegov profil. Učenici dobijaju analizu svog profila stilova učenja sa preporukama za personalizaciju. Njima se potom prikazuje varijanta sadržaja za učenje koja odgovara utvrđenom stilu učenja.

Na slici 11 prikazan je korisnički interfejs sistema *iWeaver* (Wolf, 2007). Učeniku su ponuđene osnovne korisničke opcije kao i stablo za navigaciju kroz sadržaj. Centralni deo interfejsa je namenjen za prikazivanje sadržaja. Sistem bira najadekvatniji način za prikaz sadržaja trenutnom učeniku na osnovu njegovog stila učenja. Prečice za prikaz ostalih stilova se postavljaju u *toolbar*-u za izbor stilova (varijante prikaza sadržaja). Ponuđeni su sledeći stilovi:

- **Tekstualna lekcija.** Predstavlja najjednostavniju varijantu prikaza sadržaja namenjenu učenicima koji najbolje usvajaju materijal tako što ga jednostavno čitaju. Ovaj stil sadrži formatirani tekst sa naglašenim delovima izvornog koda i podvučene ključne koncepte.
- **Animacije.** Učenicima kojima najviše odgovara vizuelno učenje prezentuje se odgovarajuća animacija koja prikazuje pojedine koncepte. Na taj način učenici kreiraju u sebi sliku na osnovu toga šta vide ili čuju.
- **Interaktivni sadržaj.** Ova varijanta sadržaja namenjena je učenicima koji bolje usvajaju gradivo pomoću aktivnog stila učenja. Sastoji se od animacija sa kojima učenik u procesu učenja uspostavlja interakciju. Na primeru *switch* naredbe, učenik može ručno postavljati *break* izraz tako što će manuelno podešavati položaj naredbe u okviru koda.
- **Audio sadržaj.** Kategorija učenika koji gradivo najbolje usvajaju kroz slušanje sadržaja lekcije nazivaju se auditorni učenici. Ova varijanta prikaza sadržaja prezentuje materijal u audio formatu, slično prikazu snimljene PowerPoint prezentacije.

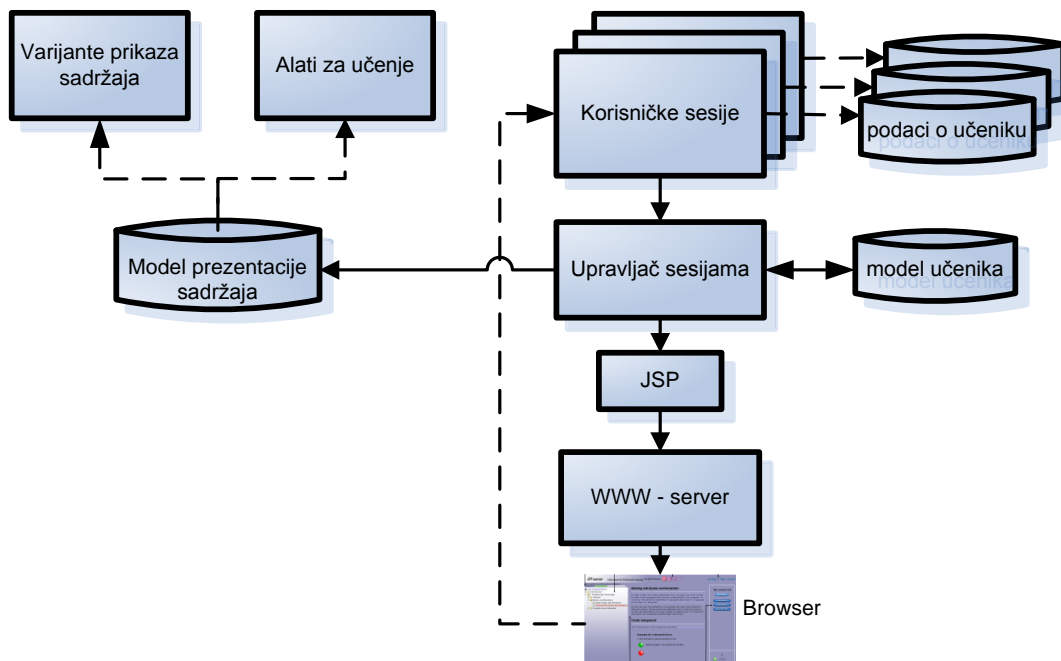


Slika 11. Korisnički interfejs sistema *iWeaver*

Sadržaj koji se uči je podeljen u module (eng. *modules*) koji su dalje podeljeni na jedinice (eng. *units*). Nakon što pređe neku od jedinica, učenik može da odabere da je ponovi ali uz upotrebu drugog stila. Ukoliko učenik odabere prelazak na narednu jedinicu, od njega se traži da oceni varijantu prikaza prethodno pređenog nastavnog sadržaja na osnovu njegovog subjektivnog mišljenja o efikasnosti. Takođe se od učenika traži da oceni svoj napredak pri učenju i opšti utisak o okruženju za učenje.

**Arhitektura sistema** - Šematska arhitektura sistema *iWeaver* (Wolf, 2007) je predstavljena na slici 12. Učenik se prijavljuje na sistem preko standardnog veb brovsra. Na taj način se pokreće učenikova sesija koja čuva sve relevantne podatke (trenutni stil prikaza, opcije navigacije, upotreba alata, itd).

Upravljač sesijama (eng. *session manager*) je realizovan kao servis koji neprekidno sakuplja podatke kroz sesije učenika. Model učenika se analizira i upoređuje sa modelom prezentacije sadržaja (eng. *Content representation model*). Zatim se pronalazi najbolja kombinacija prikaza sadržaja i alata koja se preporučuje učeniku. Ova se kombinacija zatim prikazuje kao JSP – generisana HTML stranica u učenikovom pretraživaču.



Slika 12. Arhitektura sistema *iWeaver*

Model učenika koji se koristi u sistemu *iWeaver* definiše stil učenja svakog učenika. Ne uzimaju se u obzir drugi faktori kao što su prethodno znanje i ciljevi učenja tako da su ciljna grupa ovog sistema učenici bez prethodnog znanja iz objektno-orijentisanog programiranja.

Sistem *iWeaver* primenjuje tehnike personalizacije nastavnog materijala bliske tehnikama koje se koriste u sistemu Protus, međutim sistemu nedostaje upotreba tehnologija semantičkog veća kojima bi se postigla lakša ponovna upotreba nastavnog materijala i elemenata sistema.

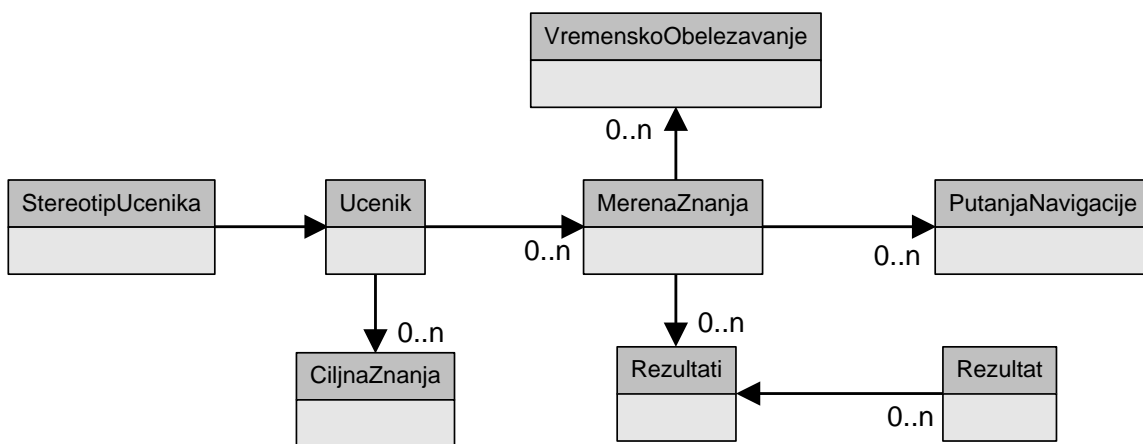
Protus 2.0 predstavlja unapređenje sistema *iWeaver* jer poseduje funkcionalnosti za dodavanje novih kurseva, nastavnog materijala kao i mogućnosti proširenja opcija personalizacije. Olakšan unos novih opcija personalizacije omogućava implementaciju

dodatnih tehnika adaptacije nastavnog materijala, kao što je generisanje preporuka na osnovu tagova kao i prilagođavanje navigacije kroz nastavni materijal.

### 5.4.2 Multitutor

*Multitutor* predstavlja onlajn okruženje za razvoj kurseva elektronskog učenja (Šimić, 2004). Dizajniran je kao klijentsko-serverski sistem koji koristi ontologije za prikaz osnovnih elemenata sistema. Sistem omogućuje predavačima da razviju tutorski sistem za pohađanje proizvoljnog kursa. Od predavača se očekuje da definiše poglavlja, lekcije i testove. Na taj način, *Multitutor* omogućuje kreiranje i pohađanje neograničenog broja kurseva. Sistem omogućuje učenicima da pohađaju različite kurseve, testira njihovo znanje nakon faze upoznavanja sa nastavnim materijalom i generiše preporuku materijala u cilju postizanja boljih rezultata. Generisanje preporuka se vrši na osnovu istorije pristupa i prethodnih rezultata učenika koji se čuvaju u modelu učenika.

Primer ontologije modela učenika sistema *Multitutor* dat je na slici 13 (Šimić, 2004). Klasa *učenika* sadrži objektivne informacije o učeniku. Kada se učenik registruje u sistemu da bi pohađao kurs, od njega/nje se zahteva da popuni upitnik čime se omogućuje sistemu da postavi inicijalne vrednosti u model učenika. Klasa *CiljnaZnanja* povezuje delove ontologije koji ne zavise od domena sa samim ontologijama domena. Osim toga sistem dozvoljava da učenik pohađa više kurseva preko Multitutora, od kojih svaki poseduje svoju ontologiju domena. Kao rezultat, model učenika sadrži početne slojevite podatke u klasi *CiljnaZnanja* i oni su kategorisani u vidu stereotipa definisanih u klasi *StereotipUčenika*. Tokom vremena, aplikacija ažurira *CiljnaZnanja* i time procenjuje nivo učenikovog znanja. Konkretno vrednosti se čuvaju u klasi *MerenaZnanja*, i svaka od njih je predstavljena kao skup *Rezultata*. Tokom interakcije učenika sa sistemom, aplikacija prati učenikovu navigaciju i vreme provedeno na učenju svakog pojedinačnog koncepta. Ti se podaci zatim zajedno koriste sa ocenama učenika u cilju adaptivne personalizacije narednog nastavnog materijala koji će biti prikazan učeniku.



Slika 13. Primer ontologije modela učenika Multitutora

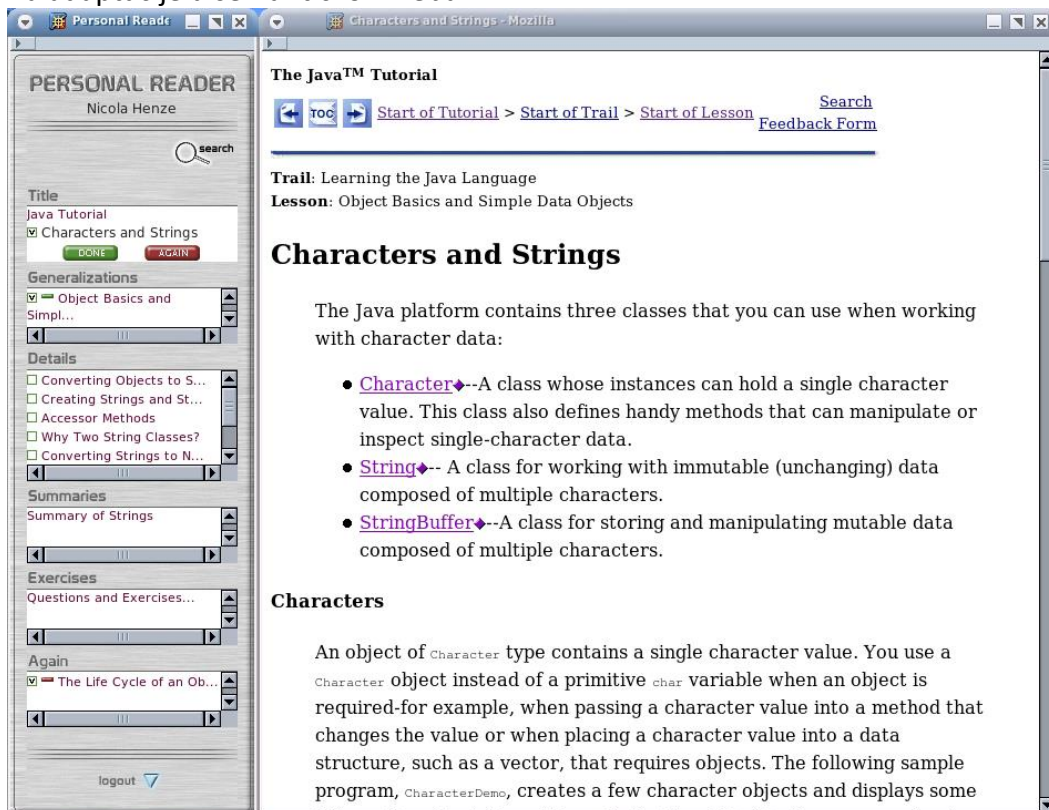
*Multitutor* predstavlja uspešnu realizaciju okruženja za elektronsko učenje, međutim iz realizacije su izostavljene brojne potencijalne mogućnosti personalizacije nastavnog materijala. Pored opširnije strukture ontologija sistema koje omogućavaju lak unos novih opcija personalizacije nastavnog materijala, sistem Protus 2.0 omogućava i laku implementaciju kuseva i nastavnog materijala iz različitih domena.

Potpunim razdvajanjem nastavnog sadržaja i njegovog dizajna sa jedne strane i aktivnosti dizajna same aplikacije i načina prezentacije materijala sa druge strane u sistemu Protus 2.0, postiže se opštost sistema kao i lakša ponovna upotreba njegovih pojedinih komponenti aplikacije.

### 5.4.3 Personal Reader

*Personal reader* predstavlja važan rezultat primene tehnologija semantičkog veba pri izgradnji sistema za elektronsko učenje (Henze et al., 2004). Sistem nudi okvir za dizajn, implementaciju i održavanje veb sadržaja, podržane opcijama personalizacije za svakog pojedinačnog učenika. Arhitektura sistema je zasnovana na jeziku RDF (eng. *Resource Description Framework*) i ontologijama za razmenu informacija o nastavnim resursima, domenu i učenicima. Resursi su definisani uz pomoć ontologija dok su upiti nad bazom i opcije adaptacije prikazani uz pomoć jezika TRIPLE.

Dizajn ovog personalizovanog sistema za učenje na daljinu je prikazan u (Henze et al., 2004). Autori su koristili tehnologije semantičkog veba za automatsko generisanje hiperlinkova u sistemu (slika 14). Definisane su ontologije i metapodaci za tri tipa resursa: ontologije domena (eng. *domain ontology*), ontologije korisnika (eng. *user ontology*) i ontologije za praćenje korisničkih sesija (eng. *observation ontology*). Međutim, strategije personalizacije nastavnog materijala nisu definisane pomoću ontologija sistema. Za izgradnju sistema primenjen je pristup zasnovan na logici generisanja obrazovne hipermedije uz pomoć jezika TRIPLE, jezika za definisanje upita i pravila adaptacije u semantičkom vebu.



Slika 14. Personal Reader

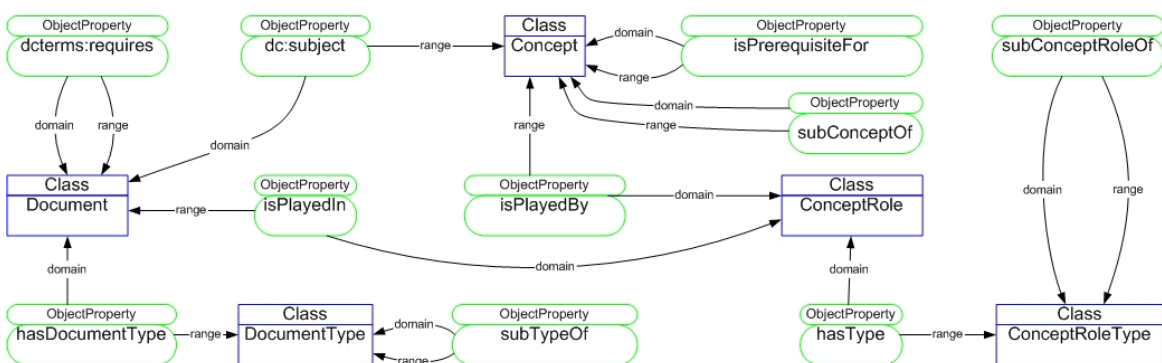
Ovo rešenje personalizovanog sistema sadrži nastavni materijal definisan metapodacima u okviru ontologije domena. U cilju personalizacije prikaza nastavnog materijala,



definisana je ontologija za modeliranje učenika i ontologija za praćenje korisničkih sesija koja prati i beleži interakciju učenika sa sistemom za elektronsko učenje. Svaki servis sistema za personalizovano učenje se sastoji od pravila adaptacije i ima jasno definisanu svrhu adaptacije.

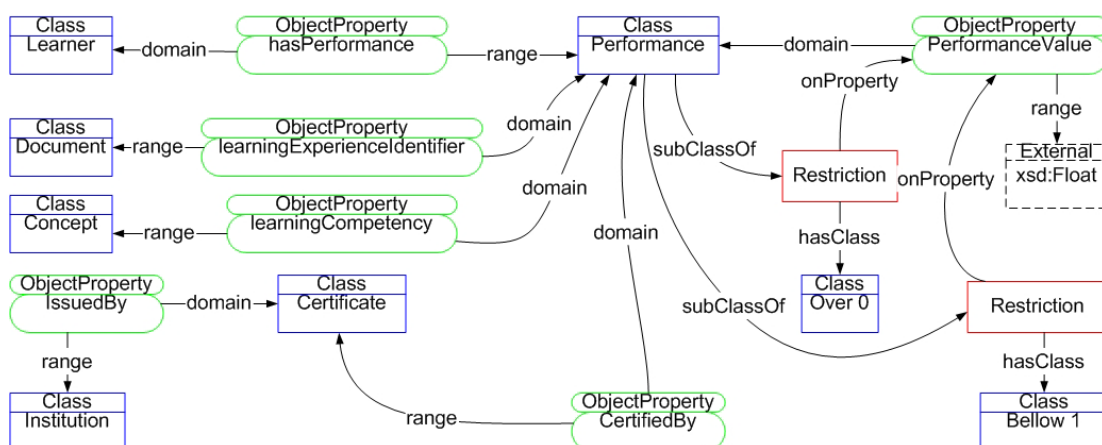
*Personal reader* generiše opcije personalizacije iz oblasti adaptivne obrazovne hipermedije (eng. *adaptive educational hypermedia*). U cilju pretrage i pronalaska informacija autori su definisali niz IF-THEN pravila za generisanje preporuka u veb kursu.

Pristup prikazan u (Henze et al., 2004) malo više pažnje dodeljuje dizajnu personalizovanog sistema za elektronsko učenje uz upotrebu algoritama za personalizaciju na način da se ti algoritmi mogu ponovno koristiti, kombinovati i po potrebi primenjivati. Sistem takođe sadrži ontologiju domena (u radu nazvanu ontologija dokumenata) prilagođenu hipermedijalnom okuženju kao skup prikaza dokumenata i koncepata u sistemu (slika 15).



Slika 15. Ontologija dokumenata sistema *Personal Reader*

Klasa *Dokument* se koristi za definisanje koncepata, tj. lekcija. Svaka lekcija sadrži više nastavnih resursa (tekstualnih ili grafičkih fajlova). Ontologija dokumenata je kreirana da naglasi funkciju nastavnih resursa u sistemu. Uz pomoć ontologije dokumenata, definiše se raspored prikaza dokumenata kao i uloga pojedinih koncepata u nastavnom procesu. Takođe, definisana je i hijerarhija koncepata, dokumenata i uloga koncepata.

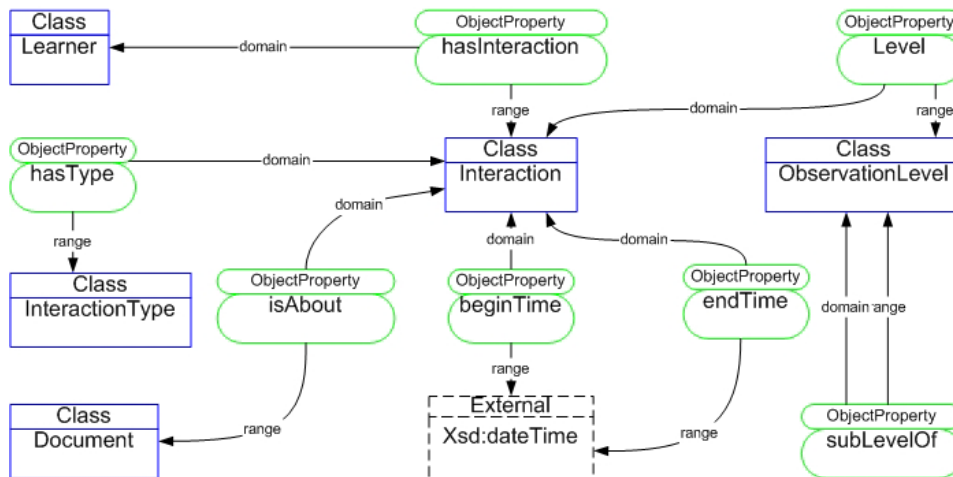


Slika 16. Ontologija profila učenika sistema *Personal Reader*

Na slici 16 je prikazana ontologija profila učenika definisanog u sistemu *Personal reader*. Ona prikazuje vezu klase učenika (klasa *Learner*) i klase za praćenje napretka učenika (klasa *Performance*). Sistem prati pristup učenika određenim dokumentima i konceptima, čime se učenici usmeravaju na nove (neposećene) koncepte i dokumente. Nedostatak

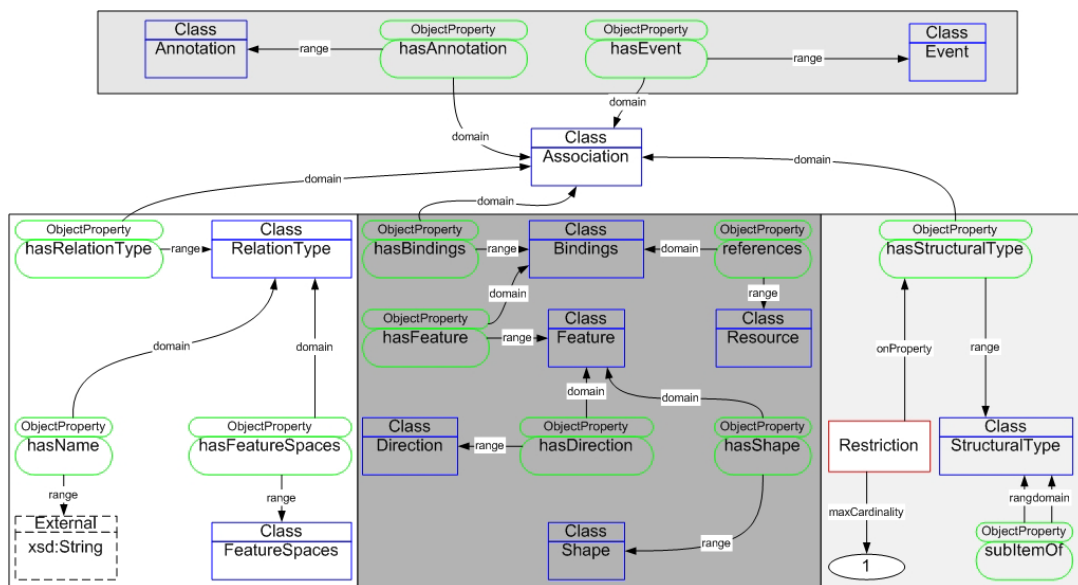
sistema je što se modeluje samo minimalan broj osobina učenika čime je onemogućena složenija personalizacija u sistemu.

Ontologija za praćenje korisničkih sesija (eng. *ontology for observations*) su prikazane na slici 17. Ontologija sadrži podatke o interakciji učenika sa odgovarajućim dokumentom. Primeri interakcije koji se beleže su pristup (eng. *access*) nastavnom materijalu, postavljanje markera (eng. *bookmark*), obeležavanje (eng. *annotate*) nastavnog materijala, rešavanje zadataka (eng. *exercise*), itd.



Slika 17. Ontologija za praćenje korisničkih sesija

Autori su predstavili i ontologiju koja se koristi za opis strukture elemenata neophodnih za vizualizaciju (slika 18) kao i sisteme pravila koji se koriste za postavljanje upita nad informacijama sadržanim u ontologijama (prikazanim u jeziku TRIPLE). Pravila se koriste prilikom donošenja odluka o prikazu nastavnog materijala.



Slika 18. Ontologija strukture elemenata sistema

*Personal reader* generiše personalizovano korisničko okruženje za prikaz nastavnog materijala, ali se sva personalizacija obavlja na osnovu praćenja poseta nastavnim resursima a ne koristi se ni jedan složeniji mehanizam za generisanje preporuka niti personalizacije kao što je identifikacija stilova učenja ili preporuke na osnovu tagova. *Personal reader* koristi nastavne resurse sa besplatno dostupnog tutorijala (*Sun Java*

Tutorijal), stoga nedostaje posebno razvijeni nastavni material prilagođen različitim potrebama raznih učenika.

Uporedni prikaz karakteristika predstavljenih sistema i sistema Protus 2.0, prikazan je u tabeli 2. Deo sistema Protus 2.0 namenjenog administratorima sistema, nudi mogućnost unosa nastavnog materijala posebno odabranog, definisanog i dizajniranog za prikaz lekcija u okviru tutorskog sistema. Na taj način, Protus 2.0 unapređuje nastavni materijal prisutan u sistemu.

**Tabela 2. Uporedni prikaz karakteristika sistema**

<b>Karakteristike sistema</b>	<b>iWeaver</b>	<b>Multitutor</b>	<b>Personal reader</b>	<b>Protus 2.0</b>
<b>Tehnike personalizacije</b>				
Adaptacija linkova	X		X	X
Adaptacija na različite stilove učenja	X			X
Integrirani sistemi za generisanje preporuka		X		X
Adaptacija na osnovu navigacije		X		X
<b>Tehnologije semantičkog vebe</b>				
Ontologija znanja iz domena		X	X	X
Ontologija modela učenika		X	X	X
Ontologija aktivnosti			X	X
Ontologija nastavne strategije		X		X
Ontologija korisničkog interfejsa			X	X
Pravila adaptacije			X	X
<b>Mogućnosti proširenja</b>				
Mogućnost dodavanja novog nastavnog materijala	X	X	X	X
Mogućnost dodavanja novih kurseva		X		X
Mogućnosti proširenja opcija personalizacije			X	X

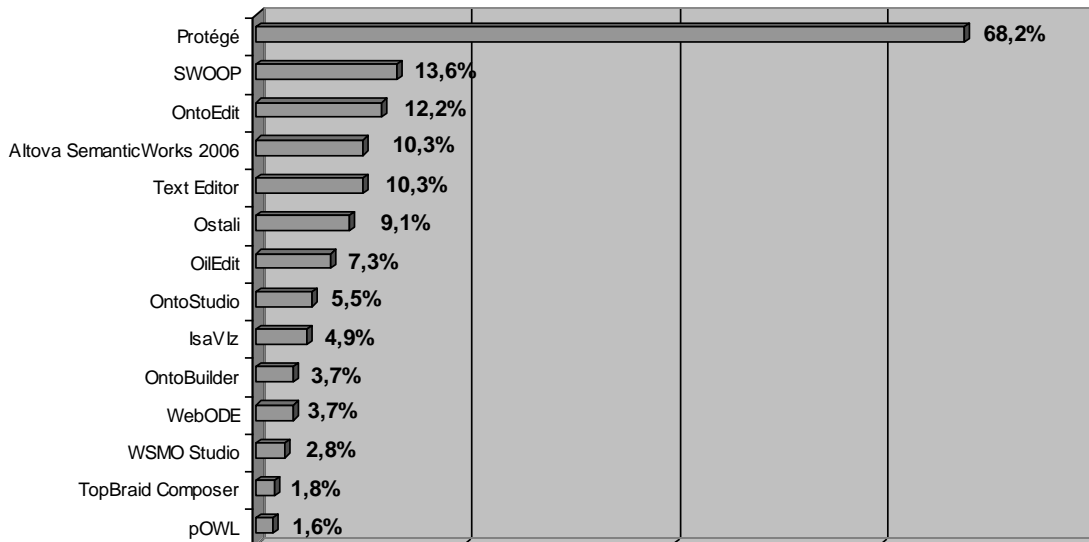
Sistem Protus 2.0 se sastoji od jasno definisanih i odvojenih komponenti koje se po potrebi mogu prilagođavati i dopunjavati. Moguća je i laka izmena opcija i mogućnosti personalizacije nastavnog materijala koji se nude učenicima uz pomoć eksplicitno definisanih i dopunjivih pravila adaptacije. Formalno definisane ontologije će omogućiti ponovnu upotrebu komponenti sistema Protus prilikom izgradnje sličnih sistema.

## 5.5 Popularnost tehnologija semantičkog vebe

Krajnja vizija semantičkog vebe tek treba u potpunosti da bude postignuta u narednom periodu, ali uočljiv je značajan progres u razvoju i upotrebi standarda, jezika, tehnologija i aplikacija. Iako će masovna primena semantičkog vebe nastupiti za najmanje desetak godina, postoji potreba za raznim vrstama ispitivanja koja će dati jasniju sliku o pravcima razvoja (Breslin, Passant, & Vrandec, 2011). U nastavku će biti prikazani osnovni trendovi razvoja semantičkog vebe, na osnovu rezultata istraživanja o upotrebi alata, sistema, jezika i tehnika. Istraživanje je sprovedeno na Univerzitetu u Madeiri, Portugalija i rezultati su predstavljeni u (Cardoso, 2007). Istraživanje je obuhvatilo 627 ispitanika (sa univeziteta i iz privrede) koji su odgovarali na 14 pitanja u vezi upotrebe tehnologija semantičkog vebe. U ovom radu će biti prikazani neki od dobijenih rezultata.

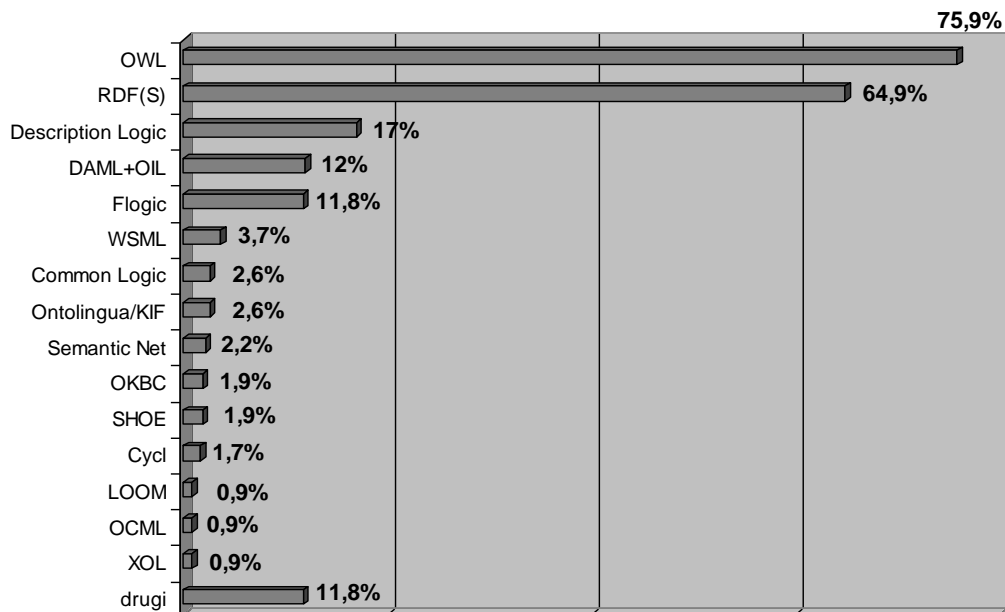
### 5.5.1 Alati i jezici za razvoj sistema baziranih na tehnologijama semantičkog veba

Ispitanici su upitani koje editore ontologija trenutno koriste u svojoj organizaciji. Rezultati prikazani na slici 19. jasno pokazuju da je *Protégé* daleko popularniji od svih drugih editora ontologija. Treba napomenuti da čak 10 procenata ispitanika koristi obične text editore za kreiranje ontologija.



Slika 19. Upotreba editora ontologija

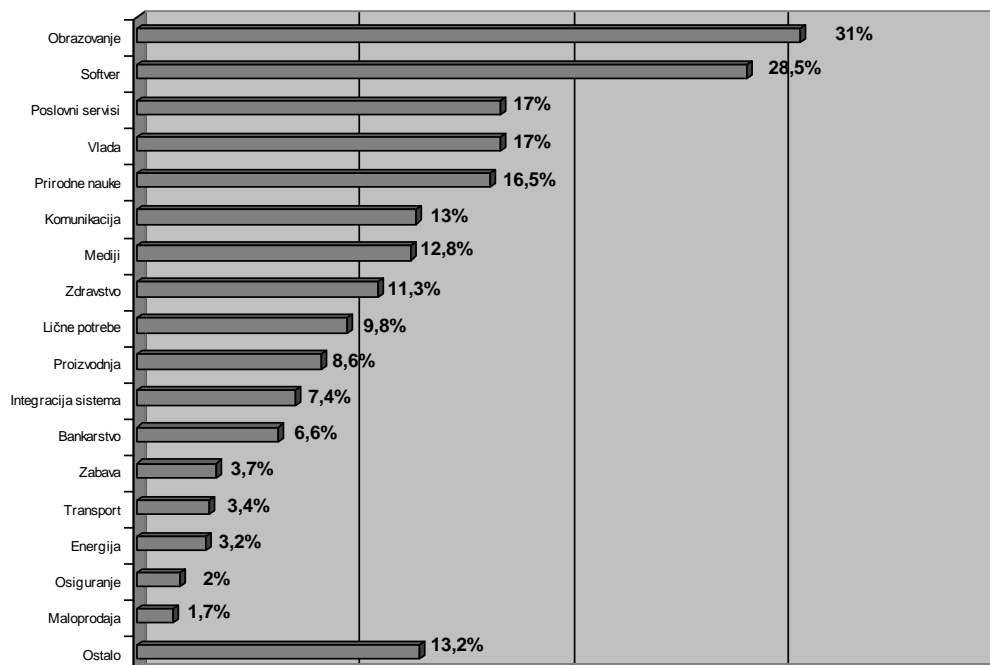
U poslednjih desetak godina razvijeno je više jezika za kreiranje ontologija (Chandrasekara et al., 1999). Odgovori su potvrdili očekivane rezultati da su daleko ispred ostalih po popularnosti OWL (eng. *Web Ontology Language*) i RDFs (eng. *Resource Description Framework*) (slika 20).



Slika 20. Ontološki jezici trenutno u upotrebi

### 5.5.2 Oblasti upotrebe ontologija

U cilju utvrđivanja trenutnih trendova razvoja ontologija za određene domene, ispitanici su upitani za koju oblast industrije oni predstavljaju znanje uz pomoć ontologija. Kao rezultat se javio veliki broj oblasti od kojih su dominantni: obrazovanje i kompjuterski softver (slika 21).



Slika 21. Industrija ili domen za koji ispitanici razvijaju ontologije

Budući da OWL predstavlja unapređenje RDF jezika a popularnost im je na sličnom nivou, na osnovu ovog istraživanja zaključuje se da semantički veb ne zahteva obavezno upotrebu OWL-a i da se mnogi ciljevi kao što su razmena podataka i integracija podataka mogu postići samo uz pomoć RDF-a kao i da najveći potencijal tehnologije semantičkog veba imaju u oblasti obrazovanja.

## Poglavlje VI

### IZGRADNJA MODELA TUTORSKOG SISTEMA

U ovom poglavlju je prikazan model opšte arhitekture tutorskog sistema koji se može koristiti za izgradnju kurseva iz različitih domena. Svi elementi tutorskog sistema su modelirani i implementirani upotrebom tehnologija semantičkog veba. Prikazan je okvir (eng. *framework*) tutorskog sistema sa opcijama za izgradnju, organizaciju i ažuriranje specifičnih obrazovnih resursa (npr. nastavnog materijala, profila učenika, putanja kroz nastavni materijal, itd).

Model tutorskog sistema definisan u disertaciji omogućuje implementaciju neograničenog broja personalizovanih kurseva iz različitih domena kao i definisanje formalnih pravila adaptacije nastavnog materijala svakom pojedinačnom učeniku. Model sadrži definicije osnovnih koncepata ontološke arhitekture i specifikaciju relacija između definisanih koncepata.

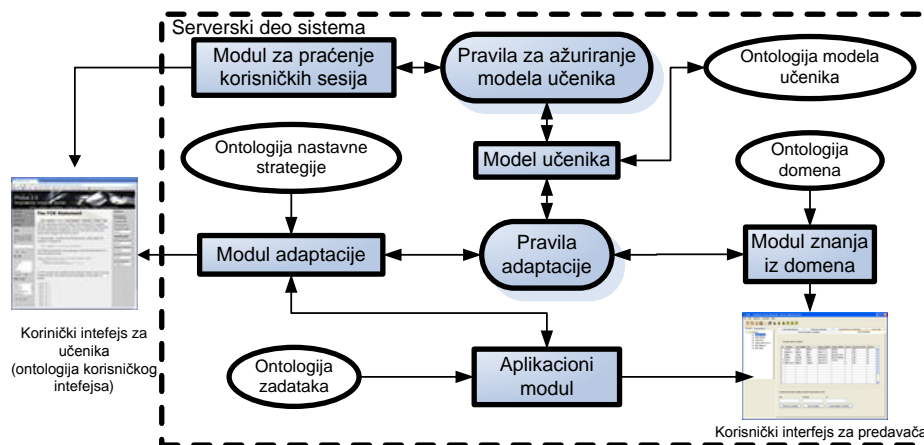
Cilj ovog poglavlja je definisanje opšteg modela tutorskog sistema, prikaz funkcionalnosti sistema i njegove arhitekture. Prikazana arhitektura uključuje upotrebu ontologija, tako što je svaka komponenta sistema predstavljena posebnom ontologijom. Na taj način se postiže jasna razdvojenost komponenti tutorskog sistema, eksplicitno se definiše komunikacija između komponenti i omogućena je upotreba naprednih tehnologija semantičkog veba za razvoj tutorskog sistema. Kao dodatak ontološkoj arhitekturi sistema, definisana su pravila adaptacije koja omogućuju lakšu dopunu i izmenu opcija personalizacije.

Predstavljeni opšti model tutorskog sistema omogućuje izgradnju personalizovanog kursa iz proizvoljnog domena kao i formalizaciju nastavnog materijala i komponenti postojećeg tutorskog sistema Protus. Protus 2.0 predstavlja novu verziju tutorskog sistema Protus (Vesin et al., 2012b) izgrađenu proširenjem opšteg modela predloženog u ovom poglavlju, definisanjem nastavnog materijala za konkretan domen. Razvoj nastavnog

materijala iz oblasti programiranja u programskom jeziku Java i njegova integracija u definisani model tutorskog sistema će biti prikazan u poglavlju 7.

## 6.1 Arhitektura sistema

Osnovna namena opšteg modela tutorskog sistema je kvalitetna adaptacija nastavnog materijala prilagođena potrebama pojedinačnog učenika na osnovu njegovog znanja, potreba i želja uz pomoć integrisanih sistema za generisanje preporuka. Arhitektura opšteg modela se sastoji iz pet funkcionalnih komponenti: modul znanja iz domena (eng. *domain module*), model učenika (eng. *learner model*), aplikacioni modul (eng. *application module*), modul adaptacije (eng. *adaptation module*) i modul za praćenje korisničkih sesija (eng. *session monitor*) (slika 22).



Slika 22. Arhitektura opšteg modela tutorskog sistema

*Modul znanja iz domena* predstavlja skladište celokupnog nastavnog materijala, tutorijala i testova. Ovaj modul je predstavljen ontologijom domena i opisuje način na koji će sadržaj biti strukturiran.

*Model učenika* predstavlja kolekciju statičkih i dinamičkih podataka o učeniku. Ovi podaci se čuvaju u okviru ontologije modela učenika. Sistem koristi te informacije da bi predvideo ponašanje učenika i na osnovu njega prilagodio nastavni materijal.

*Aplikacioni modul* je zadužen za realizaciju procesa adaptacije. Tačnije, *Modul adaptacije* preko odgovarajućih pravila adaptacije prati uputstva za prilagođavanje nastavnog materijala i korisničkog interfejsa koja su definisana u *aplikacionom modulu* i ontologiji nastavne strategije. Modul adaptacije i aplikacioni modul su razdvojeni da bi se postiglo lakše dodavanje novih kategorija nastavnog materijala.

Preko *modula za praćenje korisničkih sesija*, sistem ažurira model učenika. Ovaj proces se izvršava uz pomoć pravila za ažuriranje modela učenika.

Integracija odgovarajućih obrazovnih ontologija i pravila adaptacije za svaku komponentu sistema omogućiće razvoj tutorskog sistema u potpunosti podržanog tehnologijama semantičkog veća.

Primeri integrisanih ontologija opšteg modela tutorskog sistema, njihova upotreba i definisana pravila adaptacije će biti prikazane u poglavljima 6.2 i 6.3. Prikazan je opšti model tutorskog sistema i specifikacija komponenti tutorskog sistema, forma definisanih

ontologija kao i precizna definicija pravila adaptacije koja se primenjuju u sistemu u cilju personalizacije nastavnog materijala.

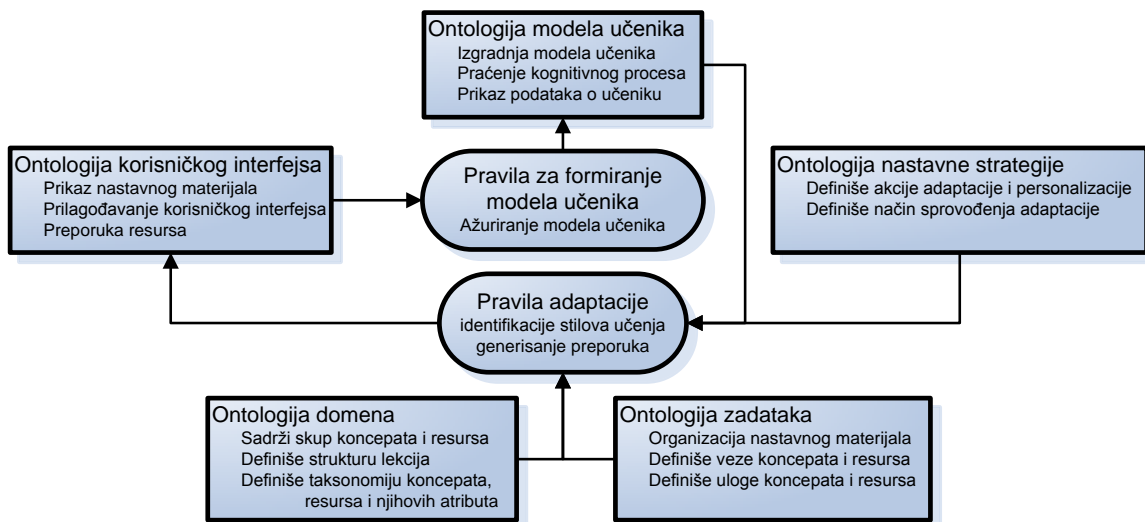
## 6.2 Ontologije sistema

Tokom poslednje decenije, posebna pažnja je usmerena na ontologije i njihovu upotrebu u aplikacijama iz oblasti obrazovanja, upravljanja znanjem, integracije podataka, itd. Inženjering ontologija predstavlja niz akcija usmeren ka razvoju ontologija posebnog domena. Na osnovu toga, inženjering ontologija predstavlja ključni aspekt unapređenja postojećeg tutorskog sistema (Strohmaier et al., 2013).

Ontologije omogućuju formalno i eksplicitno definisanje koncepata, njihovih atributa i relacija (Devedžić, 2006). Osnovni cilj disertacije je integracija sledećih obrazovnih ontologija u postojeći sistem Protus:

- ontologije za prikaz znanja iz domena (eng. *domain ontology*),
- ontologije za izgradnju modela učenika (eng. *learner model ontology*),
- ontologije za prikaz aktivnosti u sistemu (eng. *task ontology*),
- ontologije za prikaz nastavne strategije i pedagoških akcija (eng. *teaching strategy ontology*),
- ontologije za definisanje tehnika za prilagođavanje korisničkog interfejsa (eng. *interface ontology*).

Implementacija ovakvog skupa ontologija omogućava lakše deljenje znanja, njegovu ponovnu upotrebu, efikasnije modeliranje učenika i lakše proširenje sistema. Ontologije se definišu uz upotrebu fomalnog jezika za kreiranje ontologija OWL (OWL, 2012). Ontološka arhitektura prikazana u disertaciji omogućava ne samo prikaz meta-podataka nego i donošenje odluka o oblicima personalizacije i prilagođavanje nastavnog materijala potrebama pojedinačnog učenika (slika 23).



Slika 23. Ontologije i pravila adaptacije u modelu tutorskog sistema

Prilikom izrade ontologija, potrebno je koristiti otvorene, standardizovane jezike kao što su XML, RDF i OWL. Oni će omogućiti standardizaciju i formalizaciju sadržaja i omogućiti kasniju ponovnu upotrebu komponenti sistema. Stoga će u ovom poglavlju biti prikazana



arhitektura tutorskog sistema koja će omogućiti semantičku reprezentaciju komponenti i adaptivne funkcionalnosti sistema (Vesin et al. 2012b).

### 6.2.1 Osnovne komponente ontologija

Ontologija predstavlja formalni eksplicitni opis koncepata u okviru domena, njihovih atributa i ograničenja. Osnovne komponente ontologija su (Wand, Storey, & Weber, 1999):

- **Klase** (eng. *class, type, sort, category, kind*). Klase predstavljaju kolekcije objekata, tj. koncepata.
- **Instance klase** (eng. *individuals, instances*). Instance klase predstavljaju konkretne objekte nastale od klase.
- **Atributi** (eng. *attributes, properties, slots*). Atribut predstavlja direktnu binarnu relaciju koja definiše karakteristike klase. On predstavlja osobinu instance i može biti u formi konkretne vrednosti (eng. *Datatype property*) ili linka ka drugim instancama (eng. *Object property*). U jeziku OWL postoji i treći tip atributa - *Annotation property*. Ovaj atribut se koristi za definisanje dodatnih informacija (metapodataka) o klasama, instancama ili atributima objekata/tipova podataka. OWL dozvoljava dodatno obeležavanje klasa, atributa, instance klase i samih ontologija. Ti atributi mogu sadržati podatke o datumu kreiranja, autoru, komentare, reference ka resursima, itd.
- **Relacije** (eng. *relations*). Relacije predstavljaju način na koji su pojedini objekti u vezi sa drugim objektima.

Klase i skup pojedinačnih instanci klasa čine bazu znanja ontologije.

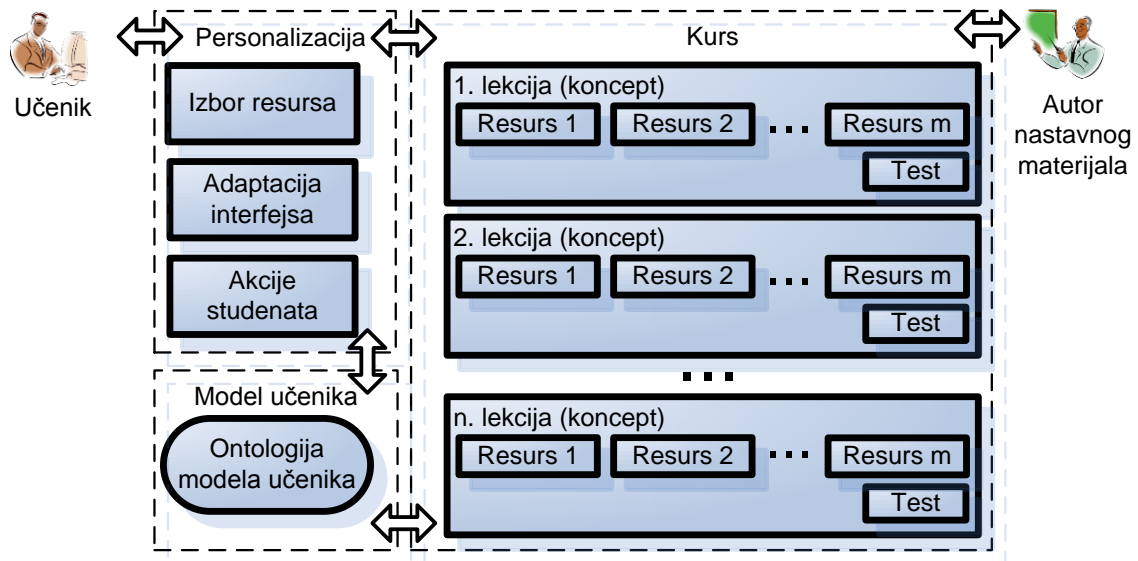
### 6.2.2 Ontologija znanja iz domena (eng. *Domain ontology*)

Važan element procesa učenja je dostupnost i razumevanje nastavnog materijala iz odgovarajućeg domena (Brusilovsky, 2003). Modul znanja iz domena predstavlja skladište celokupnog nastavnog materijala, tutorijala i testova. Ovaj modul daje opis strukture lekcija u vidu taksonomije koncepata, njihovih atributa i relacija sa drugim konceptima. Kao posledica toga javlja se prirodna ideja upotrebe ontologija za definisanje znanja iz konkretnog domena. Ontologija znanja za konkretan domen u kom se tutorski sistem primenjuje, sadrži (Vesin et al. 2012b):

- **Taksonomiju nastavnog materijala** – čine je elementi koji definišu tipove i namenu nastavnog materijala. Tako da opšti model tutorskog sistema nudi mogućnost unosa nastavnog materijala za prikaz teorijskih objašnjenja, dodatnog materijala, materijala za testiranje, itd.
- **Znanje iz domena** – čine je elementi koji predstavljaju konkretne koncepte (lekcije). Ovi koncepti mogu biti podeljeni u više kategorija i specifični su za svaki pojedinačni kurs razvijen uz pomoć definisanog opšteg modela tutorskog sistema. Na primer, konkretnom primenom opšteg modela tutorskog sistema prilikom izgradnje sistema Protus 2.0 formirani su sledeći koncepti: prikaz osnovne sintakse, naredbe ponavljanja, naredbe grananja, definisanje metoda, itd.

Opšti model tutorskog sistema prikazan u disertaciji omogućuje razvoj kurseva iz različitih domena. Svaki kurs se sastoji iz niza lekcija (koncepata) koje prikazuju pojedinačne segmente domena koji se obrađuje (slika 24). Svaka lekcija se sastoji od niza resursa koji

predstavljaju fajlove sa opisima pojedinačnih delova lekcija (uvodna razmatranja, objašnjenja, primeri, zadaci za vežbu, itd.). Za svaku lekciju se vezuje jedan ili više odgovarajućih testova za proveru znanja učenika. Na osnovu rezultata testova, utvrđuje se nivo napretka učenika, ažurira model učenika i generišu dalje opcije personalizacije.



Slika 24. Nastavni materijal u sistemu Protus 2.0

### Koncepti (eng. Concepts)

Koncept predstavlja klasu ontologije za čuvanje podataka o pojedinačnoj lekciji. Svakom konceptu je dodeljen veći broj resursa (eng. *resources*) u obliku tekstualnih fajlova, slika, ilustracija, grafikona, itd. Svaki resurs je definisan svojim tipom resursa (eng. *Resource type*): teorija (eng. *theory*), primeri (eng. *example*), zadaci (eng. *assignment*), vežba (eng. *exercise*), objašnjenja (eng. *explanation*), itd. Resursi se u definisanom opštem modelu tutorskog sistema čuvaju u obliku *html* dokumenata.

Atributi klasa (eng. *Datatype property*) *Concept* i *Resource* su prikazani u tabeli 3 (prilog 1 disertacije sadrži kompletan prikaz svih klasa ontologija sistema i njihovih atributa). Ostali podaci u ontologiji se čuvaju u formi relacija sa odgovarajućim klasama (eng. *Object property*) što će biti prikazano u poglavlju 6.2.3 u okviru prikaza ontologije zadataka (eng. *Task ontology*).

Tabela 3. Atributi modela učenika u klasama *Concept* i *Resource*

<b>Concept</b>	
hasId[type:int]	Identifikacioni broj koncepta
hasName[type:string]	Naziv koncepta
<b>Resource</b>	
hasId[type:int]	Identifikacioni broj resursa
hasName[type:string]	Naziv resursa
supports[type: string]	Uloga resursa
isVisited[type: string]	Da li je resurs posećen
isRecommended[type: string]	Da li je resurs preporučen
hasFileType[type: string]	Tip fajla
hasRole[type: string]	Uloga resursa

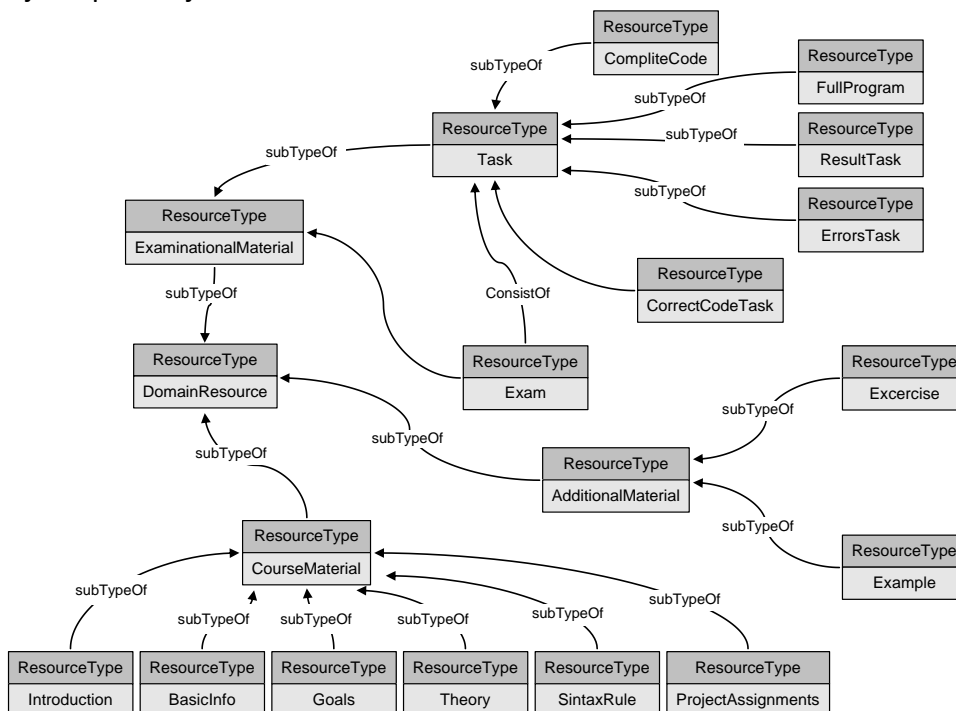
## Resursi (eng. Resources)

Svi koncepti sadrže veći broj resursa različitih tipova. Tipovi resursa se definišu u ontologiji resursa koja predstavlja deo ontologije domena. Tokom sesija učenja, sistem bira koji od ponuđenih resursa će da prikaže učeniku u zavisnosti od generisanih preporuka.

Podaci o konkretnim resursima se čuvaju u instancama klase *Resource*. Svaka instanca te klase sadrži osnovne podatke o pojedinačnom resursu koji se koristi prilikom prikaza nastavnog materijala (tabela 3).

Svi resursi se grupišu po svom tipu, ulozi i konceptu kojem su dodeljeni i ta kategorizacija predstavlja osnovu uspešne preporuke resursa tokom procesa personalizacije.

Svakom resursu se dodeljuje odgovarajući tip resursa (eng. *ResourceType*). Topologija *Resource type* ontologije je prikazana na slici 25. Ova ontologija sadrži tipove resursa koji su potrebni za odvijanje kursa iz proizvoljnog domena. U slučaju razvoja kursa iz domena koji zahteva neke specifične resurse moguće je i njih definisati. Na primer, resursi tipa *SyntaxRule* su specifični za domen programiranja. Najopštiji tip resursa je *DomainResource* (Vesin et al. 2012b). *DomainResource* poseduje tri podtipa: *CourseMaterial*, *AdditionalMaterial* i *ExaminationMaterial*. Klase *CourseMaterial* i *AdditionalMaterial* predstavljaju respektivno teoretska i praktična objašnjenja trenutne lekcije koja se prikazuje učenicima.



Slika 25. Topologija ontologije resursa opšteg modela tutorskog sistema

*ExaminationMaterial* u modelu tutorskog sistema se dodatno može podeliti na dve podkategorije: zadatke (eng. *Task*) i testove (eng. *Exam*). Test se sastoji od više zadataka. Budući da je prvi razvijen nastavni materijal u sistemu bio za kurs iz učenja programiranja, većina tipova zadataka su predviđeni za takav tip kursa. Definisani tipovi zadataka uključuju zadatke za odabir ispravnog koda, dopunu koda, ispravljanje koda, pronalaženje grešaka, itd. Prilikom razvoja kurseva iz drugih domena (koji nisu programerski) moguće je definisati zadatke opšteg tipa (eng. *General*).

Celokupni nastavni materijal određenog kursa (klasa *CourseMaterial*) se može dalje deliti u manje nastavne jedinice. Tako se nastavni materijal može podeliti na materijal tipa: uvod (klasa *Introduction*), osnovni podaci (klasa *BasicInfo*), ciljevi (klasa *Goals*), teorija (klasa *Theory*), objašnjenja (klasa *Explanation*) i definisani zadaci projekta (klasa *ProjectAssignments*). Svi navedeni tipovi materijala odgovaraju osnovnim tipovima nastavnog materijala kursa programiranja ali se mogu koristiti i za formiranje kursa iz opštog domena. Takođe se mogu definisati i novi tipovi materijala.

Ontologija prikazana na slici 25 pruža informacije potrebne za funkcionisanje ontologije zadataka (eng. *Task ontology*) i ontologiji nastavne strategije (eng. *Teaching strategy ontology*) koje će biti pojašnjene u poglavljima 6.2.3 i 6.2.5.

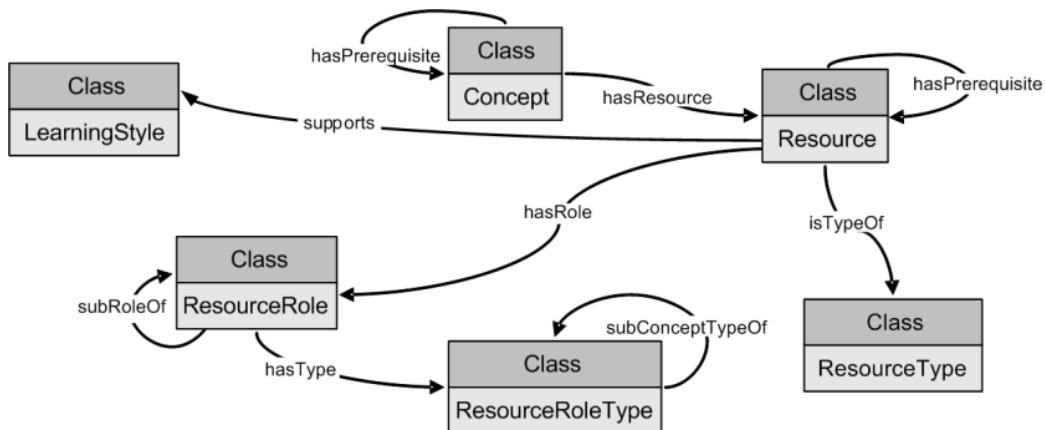
### 6.2.3 Ontologija zadataka (eng. *Task ontology*)

Ontologija zadataka opisuje strukturu pojedinih zadataka i ciljeva sistema (Vesin et al. 2012b). Ona predstavlja dodatak ontologiji domena tako što definiše semantiku svakog elementa domena znanja (konceptata i resursa). Ontologija zadataka dodjeljuje ulogu svakom objektu i relacijama između njih. Ovom ontologijom se definišu uloge pojedinih tipova konceptata i resursa.

Na primer, ako resurs ima ulogu: *fact* ili *definition*, tada se taj resurs koristi da poveća osnovno znanje učenika a ako je njegova uloga *example* tada on služi za povećanje praktičnih sposobnosti učenika.

Ontologija zadataka u opštem modelu tutorskog sistema je data na slici 26. Ova ontologija predstavlja organizaciju nastavnog materijala. Klasa *Concept* se koristi da se predstavi jedna nastavna jedinica tj. pojedinačna lekcija.

Svaki koncept se prikazuje učeniku uz pomoć većeg broja resursa (eng. *Resource*). Na taj način u opštem modelu tutorskog sistema svaki koncept (lekcija) sadrži: uvod, teoriju, objašnjenja, primere, zadatke, itd. Svaki od tih elemenata je prikazan odgovarajućim resursom, tj. fajlom u formi *html* dokumenta, tekstualnog fajla ili slike u *jpg* formatu.



Slika 26. Ontologija zadataka sistema opšteg modela tutorskog sistema

Klase konceptata i resursa su povezane relacijom *hasResource*.

Između dva konceptata/resursa se može uspostaviti relacija *hasPrerequisite*. Ova relacija pomaže navigaciju kroz koncepte/resurse. Ona omogućuje definisanje nekih konceptata/resursa koje učenik mora savladati pre nego što pređe na druge konkretne koncepte/resurse. Sistem u tom slučaju zna da se neki koncept/resurs ne može

predstaviti učeniku ako on nije prethodno savladao koncept/resurs koji je preduslov za koncept/resurs koji se trenutno obrađuje. SWRL pravilima se ažurira ontologija sistema i na taj način realizuje navigacija kroz nastavni materijal.

Resursi igraju određenu ulogu prilikom savlađivanja odgovarajućih koncepata (lekcija). Na primer, neki resursi predstavljaju ključne informacije dok drugi predstavljaju samo sredstvo da se ponude dodatne informacije ili poređenja.

Ove informacije u prikazanoj ontologiji su predstavljene instancama klasa *ResourceRole* i *LearningStyle*, kao i njihovim relacijama sa klasom *Resource*: *hasRole* i *supports*.

Na primer, resursi kao što su *BasicInfo* i *Example* imaju različite uloge. Uloga prvog je da prikaže uvodne informacije o lekciji dok je uloga drugog da ponudi dodatne primere. Sa druge strane, oba resursa predstavljaju podršku adaptaciji sistema na *Refleksivni* stil učenja.

Resursima se može dodeliti tip resursa (*ResourceType*). Koncepti, resursi i njihovi tipovi formiraju ontologiju zadataka u opštem modelu tutorskog sistema.

#### **6.2.4 Ontologija modela učenika (eng. *Learner model ontology*)**

Izgradnja modela učenika i praćenje kognitivnog procesa učenika su važni aspekti implementacije personalizacije u tutorskom sistemu. Model učenika predstavlja prikaz podataka o pojedinačnom učeniku važnih za proces adaptacije. Sistem koristi podatke iz modela učenika da predvidi ponašanje učenika i na osnovu toga prilagodi nastavni materijal (Vesin et al. 2012b).

Podaci u modelu učenika u opštem modelu tutorskog sistema su podeljeni u tri sloja kao što je predloženo u (Klašnja-Milićević et al. 2011b):

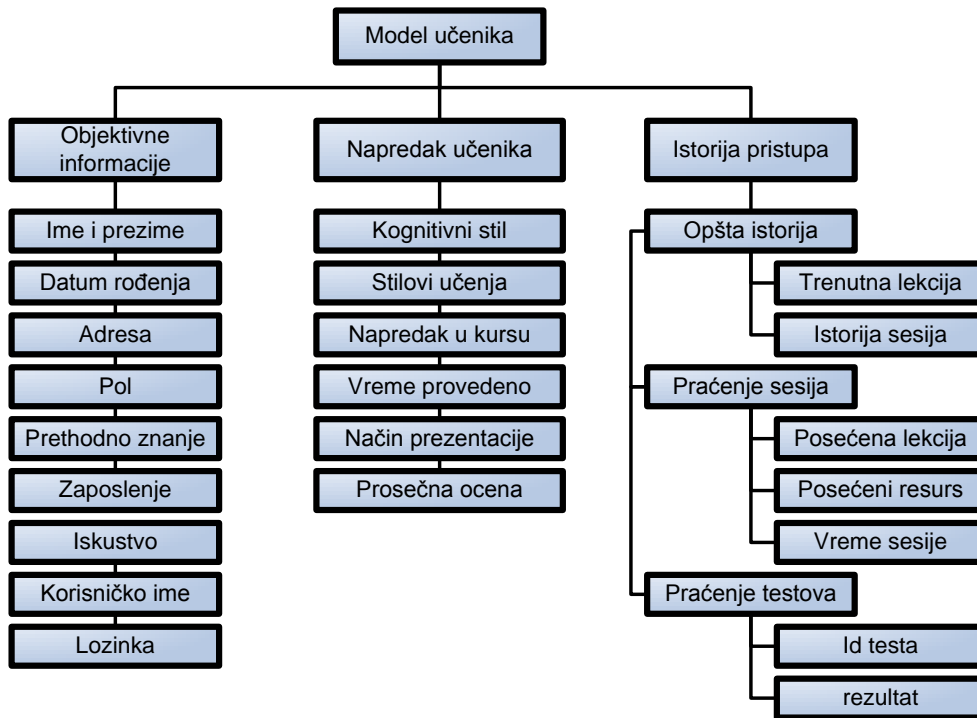
- *objektivne informacije*, koje uključuju podatke koje je učenik uneo nakon prvog pristupa sistemu kao što su: lični podaci, prethodno znanje, želje, itd.,
- *napredak učenika*, koji sadrži podatke o nivou znanja učenika, napravljenim greškama, načinjenom napretku i ukupno postignutom uspehu na testovima,
- *istorija pristupa*, koji uključuje informacije o lekcijama, konceptima, resursima i testovima koje je učenik posetio i/ili savladao, podaci o interakciji sa sistemom, rezultatima testiranja itd.

Ontologija modela učenika u opštem modelu tutorskog sistema pomaže pri sakupljanju informacija o interakciji učenika sa sistemom: korisničko ime učenika, datum i vreme pristupa kao i numeričke rezultate pristupa (posećeni resursi, rezultati testova, itd.). Ova ontologija se automatski ažurira svaki put kad neki zahtev ili akcija dospe na veb server (na primer, podacima o promeni lekcije, resursa, ocene na testu, itd.).

Sakupljene informacije se klasifikuju kroz tri sloja modela učenika koji su predstavljeni na slici 27.

Model učenika sadrži podatke o ličnim karakteristikama učenika i njegovom napretku u savlađivanju gradiva (Vesin et al. 2012a). Informacije se redovno ažuriraju tokom interakcije učenika sa nastavnim sadržajem i koriste prilikom donošenja zaključaka i odluka u ontologiji nastavne strategije (eng. *Teaching strategy ontology*).

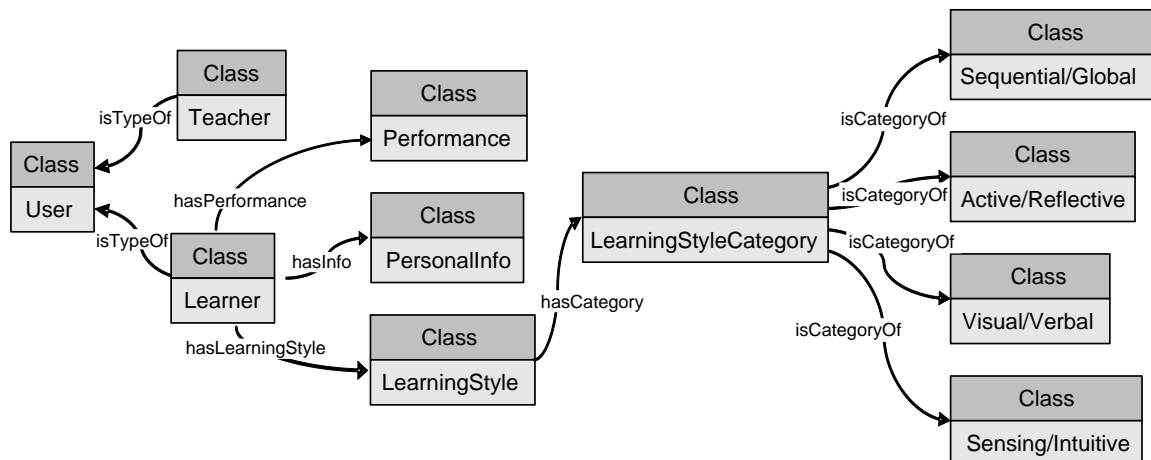
Ontologija modela učenika nudi mogućnosti prikaza svih informacija o učeniku, od poverljivih informacija kao što je lozinka do podataka o razvoju sposobnosti i znanja učenika.



Slika 27. Nivoi u modelu učenika

U modelu tutorskog sistema definisane su dve kategorije korisnika: predavači (prikazani klasom *Teacher*) i učenici (predstavljani klasom *Learner*). Klase *Teacher* i *Learner* predstavljaju podklase klase *User*.

Klasa učenika (*Learner*) se sastoji od tri komponente (klase) sa podacima o napretku učenika (*Performance*), ličnim podacima (*PersonalInfo*) i individualnim stilovima učenja (*LearningStyle*). Ove tri klase su povezane sa klasom učenika relacijama: *hasPerformance*, *hasInfo* i *hasLearningStyle* (slika 28).



Slika 28. Ontologija modela učenika opšteg modela tutorskog sistema

Klase *Learner*, *Performance*, *PersonalInfo* i *LearningStyle* predstavljaju ontologiju modela učenika. Svaka od klasa sadrži određeni broj atributa (eng. *Datatype Properties*) koji služe

za čuvanje podataka o svakom pojedinačnom učeniku. Atributi klasa *Learner*, *Performance*, *PersonalInfo* i *LearningStyle* su prikazani u tabeli 4. Ostali podaci u ontologiji se čuvaju pomoću relacija sa odgovarajućim klasama (eng. *Object property*) što će biti prikazano u nastavku ovog poglavlja. Prilog 1 disertacije sadrži kompletan prikaz svih klasa ontologija sistema i njihovih atributa.

**Tabela 4. Atributi klasa Ontologije modela učenika**

<b>Learner</b>	
id [type:int]	Identifikacioni broj učenika
<b>PersonalInfo</b>	
learner[type:int]	Identifikacioni broj učenika
name[type:string]	Ime učenika
lastname[type:string]	Prezime učenika
gender[type:string]	Pol učenika
address[type:string]	Adresa učenika
birthDate[type:dateTimeStamp]	Datum rođenja
birthPlace[type:string]	Mesto rođenja
previousKnowledge[type:string]	Prethodno znanje
affiliation[type:string]	Zanimanje
<b>Performance</b>	
learner[type:int]	Identifikacioni broj učenika
course[type:int]	Trenutni kurs
percentage[type:double]	Procenat savladanosti kursa
avgGrade[type:double]	Prosečna ocena
lesson[type:int]	Poslednja savladana lekcija
curLesson[type:int]	Trenutna lekcija
resurs[type:int]	Trenutni resurs
<b>LearningStyle</b>	
learner[type:int]	Identifikacioni broj učenika

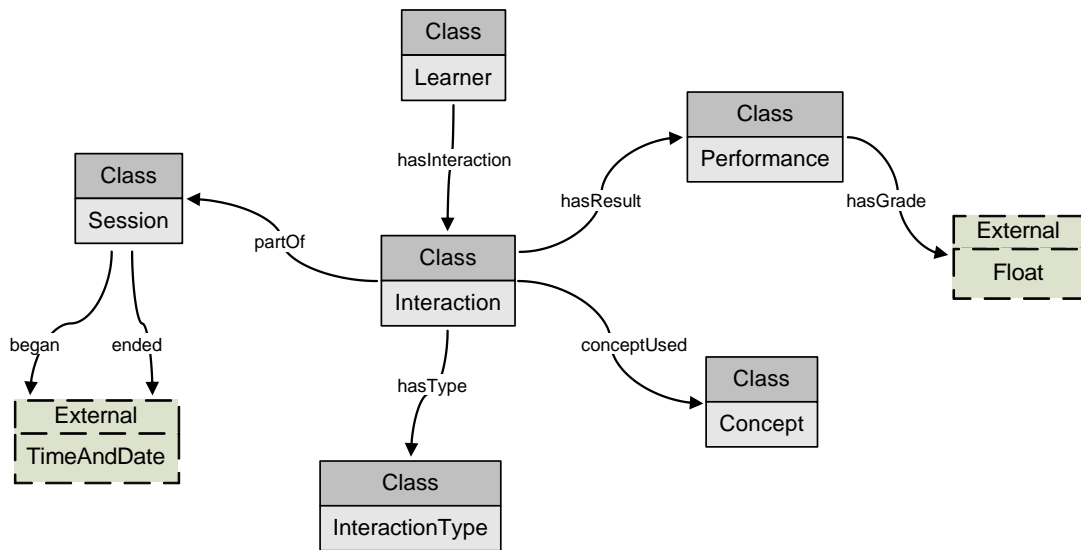
Klasa sa podacima o individualnim stilovima učenja – *LearningStyle*, sadrži podatke o stilu učenja odgovarajućeg učenika. Ova klasa nudi četiri podkategorije na osnovu dimenzija Felder-Silverman modela stilova učenja (Vesin et al. 2012a) opisanih u poglavlju 3.2.1:

- sekvencijalni/globalni,
- aktivni/refleksivni,
- vizualni/tekstualni i
- osećajni/intuitivni.

Tokom korisničkih sesija, realizuje se interakcija između učenika i sistema. Detalji o interakciji mogu pomoći prilikom zaključivanja o mogućim interesovanjima, ciljevima i znanju učenika. Ti se zaključci koriste prilikom personalizacije sistema. Stoga, ontologija mora da pruži mogućnosti praćenja i beleženja detalja o mogućoj interakciji učenika sa sistemom.

Slika 29 prikazuje ontologiju praćenja i modelovanja učenika u modelu tutorskog sistema koja predstavlja deo ontologije modela učenika. Akcije učenika se beleže u klasi za prikaz interakcije (klasa *Interaction*). Interakcija podrazumeva sve relevantne akcije koje sprovodi učenik tokom određene sesije (klasa *Session*). Interakcija uvek uključuje neki koncept, tj. relaciju sa klasom *Concept* preko relacije *conceptUsed*. Interakcija može

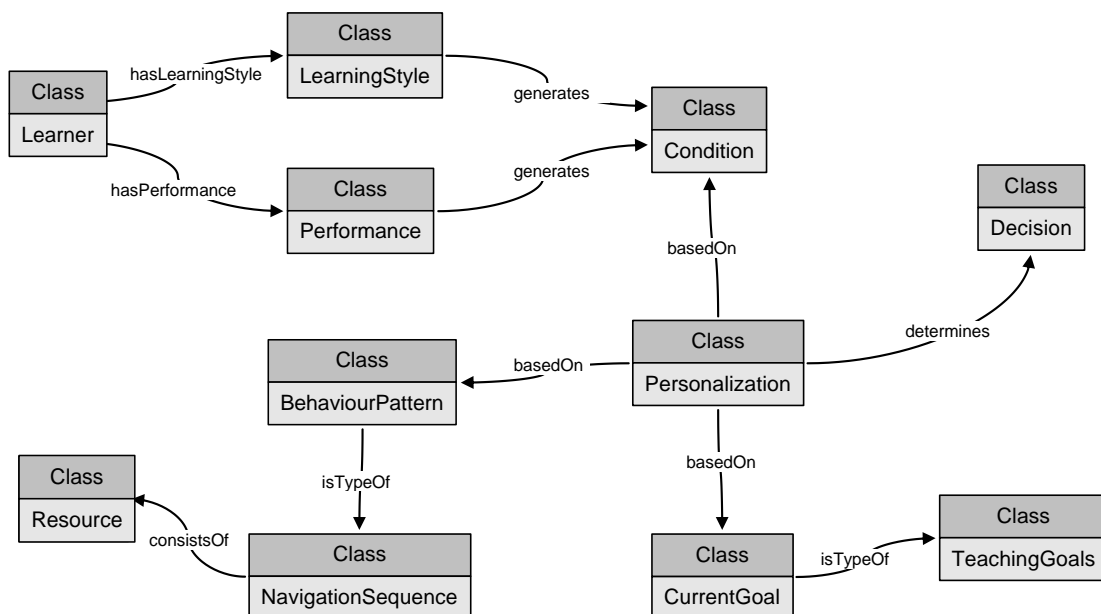
sadržati i neku numeričku vrednost koja se predstavlja u klasi *Performance*, koja je u ovom slučaju ocena u rasponu 1 do 5. Ova ontologija je zadužena za ažuriranje ontologije modela učenika.



Slika 29. Ontologija praćenja i modelovanja učenika u opštem modelu

### 6.2.5 Ontologija nastavne strategije (eng. *Teaching strategy ontology*)

Definisanje akcija adaptacije i personalizacije u tutorskom sistemu predstavlja kreiranje modela učenika i primenu različitih strategija i tehnika personalizacije u cilju efikasne adaptacije nastavnog materijala potrebama različitih učenika i njihovim individualnim stilovima učenja (Vesin et al. 2012b).



Slika 30. Ontologija nastavne strategije sistema opšteg modela tutorskog sistema

Slika 30 prikazuje način na koji se adaptacija sprovodi u *Ontologiji nastavne strategije*. Odluke se donose na osnovu informacija iz klase *Condition* (koje se generišu na osnovu informacija o stilu učenika i prethodnim rezultatima interakcije učenika sa sistemom) i ciljeva kursa. Neophodne informacije za uslove prikazane u klasi *Condition* se sakupljaju iz



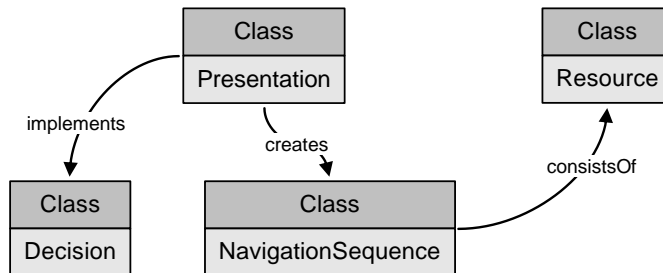
prethodno opisanih ontologija (*ontologije modela učenika, ontologije zadataka i ontologije znanja iz domena*)

Personalizacija predstavlja izbor odgovarajućeg resursa ili rasporeda njihovih prikaza koji će biti preporučeni učeniku. Akcija zavisi od više uslova ali uvek se donosi samo jedna odluka. Odluka utvrđuje koji koncept i resurs(i) će biti prikazani učeniku.

### 6.2.6 Ontologija korisničkog interfejsa (eng. *Interface ontology*)

*Ontologija korisničkog interfejsa* predstavlja poslednju fazu u komunikaciji između komponenti prikazane arhitekture i prilagođavanja korisničkog interfejsa trenutno aktivnom učeniku. Slika 31 prikazuje korake koji se sprovode u ovoj ontologiji. Ontologija čita odluke iz *Ontologije nastavne strategije* i na osnovu nje kreira niz resursa koji se preporučuju aktivnom učeniku i prikazuju na korisničkom interfejsu (Vesin et al. 2012b).

Na primer, ako klasa *Decision* uključuje određeni tip prezentacije preporučeni za datog učenika, tada se učeniku prikazuje resurs odgovarajućeg tipa. Sa druge strane, ako je u klasi *Decision* prikazan preporučeni raspored resursa tada sistem dodaje preporučeni resurs trenutnoj klasi *NavigationSequence*.



Slika 31. Ontologija korisničkog interfejsa opšteg modela tutorskog sistema

Ontologija korisničkog interfejsa se može koristiti za definisanje sadržaja stranice ili za izbor nastavnog materijala koji se prikazuje učeniku. Ovaj proces se obavlja uz pomoć pravila adaptacije i biće opisan u poglavlju 6.3.

## 6.3 Pravila adaptacije

SWRL pravila predstavljaju jedan od najpopularnijih formi prikaza znanja u sistemima semantičkog veba, zahvaljujući svojoj jednostavnosti, razumljivosti i izražajnosti (Romero et al., 2006). Postoji nekoliko različitih tipova pravila u zavisnosti od njihove svrhe i kategorije znanja koje prikazuju.

SWRL pravila se dele u nekoliko kategorija:

- pravila za donošenje odluka (eng. *decision rules*),
- pravila veze (eng. *association rules*),
- pravila klasifikacije (eng. *classification rules*),
- pravila predviđanja (eng. *prediction rules*),
- pravila za definisanje uslova (eng. *causal rules*),
- pravila optimizacije (eng. *optimization rules*), itd.

Pravila u opštem modelu tutorskog sistema se mogu podeliti na dve kategorije (Vesin et al. 2012b):

- **Pravila modelovanja učenika** (eng. *learner modeling rules*). Pravila koja pripadaju ovoj kategoriji služe za unos i ažuriranje podataka o učeniku u ontologiju modela učenika. Ova pravila su neophodna za modifikaciju inicijalnog stila učenja pojedinačnih učenika na osnovu podataka o interakciji učenika sa sistemom.
- **Pravila adaptacije** (eng. *adaptation rules*). Pravila koja pripadaju ovoj kategoriji služe za definisanje strategija adaptacije, uzimajući u obzir podatke iz ontologije znanja iz domena, ciljeva adaptacije, podataka o učeniku (iz *ontologije modela učenika*) i metoda prezentacije nastavnog materijala. Ova pravila su neophodna za generisanje adaptacije na osnovu individualnih stilova učenja (Popescu et al., 2007).

Pravila adaptacije se definišu radi postizanja ciljeva učenja uz upotrebu postojećeg nastavnog materijala a na osnovu trenutnog znanja učenika, njihovih želja i stilova učenja. Takođe, definisanje pravila adaptacije zahteva analizu postojećih metoda i tehnika adaptacija, kao što su sakrivanje/naglašavanje teksta ili linkova, izbor različitih metoda prezentacije, izmena rasporeda prikazivanja resursa, itd (Carmagnola et al., 2005).

Prednost u prikazivanju nastavnih strategija u formi SWRL pravila je što se strategije mogu eksplicitno prikazati, pregledati, menjati i ponovo koristiti u drugim sistemima.

### 6.3.1 Sintaksa pravila adaptacije

Pravila adaptacije koja su definisana u sistemu Protus 2.0 se sastoje od uslova definisanih u telu (eng. *antecedent*) i njihove posledice (eng. *consequent*) (SWRL, 2012).

Neformalno, značenje pravila je: ako su uslovi definisani u telu tačni tada moraju biti zadovoljene i definisane posledice. Tipično SWRL pravilo ima sledeći oblik:

$$a_1 \wedge a_2 \wedge \dots \wedge a_n \rightarrow b_1 \wedge b_2 \wedge \dots \wedge b_m$$

gde su  $a_i$  i  $b_i$  OWL atomi koji mogu biti:

- *Koncepti* (eng. *concepts*),  $C(x)$ , gde  $C$  predstavlja proizvoljni OWL naziv (ime koncepta), dok  $x$  predstavlja promenjivu, OWL element (činjenice o pripadnosti klasi, vrednost atributa ili naziv instance klase) ili konkretnu numeričku ili tekstualnu vrednost (eng. *data values*).
- *Atributi objekata* (eng. *Object properties*),  $P(x, y)$ , gde  $P$  predstavlja OWL atribut (eng. *property*) dok su  $x$  i  $y$  promenjive (eng. *variables*), instance klase (eng. *individuals*) ili konkretne numeričke ili tekstualne vrednosti (eng. *data values*).
- *Svojstva tipova podataka* (eng. *Datatype properties*), npr.  $P(x, y)$ , gde  $P$  predstavlja OWL atribut (eng. *property*),  $x$  predstavlja promenjivu (eng. *variable*) ili instancu klase (eng. *individual*), dok je  $y$  konkretna numerička ili tekstualna vrednost (eng. *data value*).
- $B(x_1, x_2, \dots, x_n)$ , gde je  $B$  ugrađena relacija a  $x_1, x_2, \dots, x_n$  su ili promenjive (eng. *variables*), instance klase (eng. *individuals*) ili konkretne numeričke ili tekstualne vrednosti (eng. *data values*).
- *Funkcije sameAs* ( $x, y$ ) ili *differentFrom* ( $x, y$ ) gde su  $x$  i  $y$  promenjive (eng. *variables*), instance klase (eng. *individuals*) ili konkretne numeričke ili tekstualne vrednosti (eng. *data values*).

Prazno telo predstavlja trivijalnu istinu a prazna lista posledica se tretira kao trivijalna neistina. Pravila bez tela se koriste za predstavljanje bezuslovne istine, mada je bolja praksa da se te pravilnosti definišu u samom OWL jeziku bez upotrebe pravila.

Neprazno telo i posledica pravila su tačna ako su svi njegovi atomi tačni budući da oni u stvari predstavljaju konjunkciju više atoma.

Apstraktna sintaksa SWRL jezika je prilagođena OWL specifikaciji i korisna je za definisanje XML i RDF naredbi, ali kao takva je prilično nečitka (SWRL, 2012). Stoga se u literaturi koristi relativno neformalna čitljivija forma. U toj neformalnoj sintaksi, pravila imaju oblik:

antecedent  $\Rightarrow$  consequent

ili

uslov  $\Rightarrow$  posledica

gde uslov i posledica predstavljaju konjunkcije atoma napisanih u obliku:  $a_1 \wedge \dots \wedge a_n$ .

Standardna konvencija je da se promenjive prikazuju uz pomoć svog imena kojem prethodi znak pitanja (npr. ?x).

Na sledećem primeru je prikazano pravilo napisano korišćenjem neformalne sintakse:

$\text{roditelj}(?x,?y) \wedge \text{brat}(?y,?z) \Rightarrow \text{stric}(?x,?z)$

Prethodni kod definiše pravilo: neka su  $x, y$  i  $z$  instance klase *Osoba*, i neka su u relacijama takvim da je osoba  $y$  roditelj osobe  $x$ , i da su osobe  $y$  i  $z$  braća, tada je po automatizmu osoba  $z$  stric osobe  $x$ , tj. nad tim dvema instancama klase se definiše relacija *stric*.

Formalni oblik naredbe koja sabira dve vrednosti (na primer promenjive  $x$  i broja 3, i zatim dodeljuje rezultujuću vrednost promenljivoj  $z$ ) napisane u SWRL jeziku je:

$\text{op:numeric-add}(?x,3,?z)$

Ista naredba, u neformalnoj sintaksi se prikazuje na sledeći način:

$?x = \text{op:numeric-add}(3,?z)$

Tokom nastavnih aktivnosti učenika u sistemu Protus 2.0, najvažniji zadatak predstavlja pokretanje pravila adaptacije i personalizacija korisničkog interfejsa na osnovu tih pravila. Istinitosna vrednost uslova svakog pravila se proverava i izvršava se odgovarajuća posledica ako je uslov zadovoljen.

Definisani sistem pravila u okviru opšteg modela tutorskog sistema ima opštu formu te bi mogao biti upotrebljen i u drugim sistemima i za kurseve iz drugih domena. Administratori sistema, uz pomoć *Protégé* sistema mogu menjati, brisati ili dodavati nova SWRL pravila adaptacije čime bi se omogućilo dodavanje novih opcija personalizacije u sistem. *Protégé* nudi *SWRLTab* editor za definisanje SWRL pravila adaptacije.

Tokom rada sistema, definisana SWRL pravila adaptacije se automatski pokreću uz pomoć *Jess rules engine*-a. U Java kodu sistema koristi se *Jess's Java API* koji omogućuje kreiranje instance *Jess rule engine*-a. Instanca ove klase učitava sva definisana pravila i proverava uslove definisane u njima te pokreće izvršavanje pravila ako su svi uslovi ispunjeni. Nakon pokretanja odgovarajućih pravila, ažurirani podaci se zapisuju u OWL ontologije i na taj način se ažurira baza znanja.

U nastavku rada je prikazan pristup definisanja pravila za adaptaciju učenja u opštem modelu tutorskog sistema. Definišu se tipična pravila adaptacije nastavnog materijala za prilagođavanje potrebama učenika. Prvo će biti prikazana adaptacija na osnovu identifikovanih stilova po Felder-Silverman modelu učenja, a potom pravila za generisanje preporuka i ažuriranje ontologije modela učenika.

### 6.3.2 Pravila adaptacije na osnovu identifikacije stilova učenja

Početna identifikacija individualnih stilova učenja (eng. *Learning styles identification*) u sistemu se vrši pomoću Indeksa stilova učenja - *Index of Learning Styles (ILS)* (Felder & Soloman, 1996). Na osnovu inicijalnog upitnika se procenjuju varijacije u stilovima učenja u okviru četiri dimenzije (domena). Dimenzije su:

- Obrada informacija (eng. *Information Processing*),
- Percepcija informacija (eng. *Information Perception*),
- Prijem informacija (eng. *Information Reception*) i
- Razumevanje informacija (eng. *Information Understanding*).

U okviru svake od dimenzija postoje dve kategorije učenika:

- *Obrada informacija*: Aktivni i Refleksivni učenici,
- *Percepcija informacija*: Osećajni i Intuitivni učenici,
- *Prijem informacija*: Vizuelni i Verbalni učenici,
- *Razumevanje informacija*: Sekvencijalni i Globalni učenici.

Na početku pohađanja kursa a nakon inicijalne identifikacije stila učenja, kategorija stila učenja koja je identifikovana za aktuelnog učenika mora biti zapisana u ontologiji modela učenika (eng. *Learner model ontology*) sistema (Vesin et al. 2012b).

Na primer, ako sistem odredi da učenik pripada kategoriji aktiviste iz domena *Procesiranja informacija* onda se ontologija sistema ažurira primenom sledećeg pravila:

$$\text{Learner}(?x) \wedge \text{hasLearningStyle}(?x, ?y) \wedge \text{hasCategory}(?y, ?z) \wedge \text{isCategoryOf}(?z, \text{active}) \rightarrow \text{hasLearningStyle}(?x, \text{active})$$
 PA1

Značenje pravila adaptacije PA1 je: ako je trenutno aktivnom učeniku ( $x$ ), na osnovu inicijalne identifikacije stila učenja utvrđen aktivni stil učenja ( $y$ ) tada treba ažurirati ontologiju modela učenika i tom učeniku dodeliti aktivni stil učenja (oznaka *active*). Analognim pravilima se vrši unos ostalih kategorija u okviru drugih domena stilova učenja.

Na početku svake korisničke sesije u sistemu, učitava se trenutno stanje kursa za aktivnog učenika iz ontologije modela učenika. Učitavaju se podaci o trenutnoj lekciji i kategoriji stila učenja iz svakog od četiri domena. Na osnovu tih podataka se šalje zahtev aplikacionom modulu za odgovarajući resurs koji će biti prikazan učeniku. U nastavku sesije prate se sve aktivnosti učenika kao i zahtevi koje učenik pošalje sistemu (posećene lekcije i resursi, odabrani testovi, uneti tagovi, itd).

Proces personalizacije se određuje na osnovu trenutnog stila učenja i napretka učenika. Odgovarajuće pravilo iz skupa pravila adaptacije sistema započinje personalizaciju na osnovu odgovarajućih uslova (eng. *condition*). Za učenika sa aktivnim stilom učenja, generisani uslov je *act*, za učenika sa refleksivnim stilom učenja generisani uslov je *ref*, itd.

Primeri pravila koji pokreću te akcije su:

$Learner(?x) \wedge hasLearningStyle(?x, active) \wedge Generates(active, ?z) \wedge$  PA2  
 $Condition (?z) \rightarrow Condition(act)$

$Learner(?x) \wedge hasLearningStyle(?x, reflective) \wedge$  PA3  
 $Generates(reflective, ?z) \wedge Condition (?z) \rightarrow Condition(ref)$

Značenje prethodnih pravila je: ako trenutno aktivni učenik (x) ima određeni stil učenja (aktivni/refleksivni) i taj stil generiše određeni uslov (z) tada se inicijalizuje odgovarajuća instanca klase *Condition* (aktivni stil generiše uslov *act*, refleksivni stil generiše uslov *ref*). Generisani uslovi se koriste za određivanje daljih akcija sistema koje će biti sprovedene u nastavku adaptivnog učenja.

Nakon što je definisan tip adaptacije, sistem donosi izbor odgovarajućeg nastavnog materijala koji će biti prikazan učeniku na osnovu te odluke (Vesin et al. 2012b).

Postoje tri tipa adaptacije u sistemu: *styleMatch* (prilagođavanje nastavnog materijala određenom stilu učenja), *adaptInterface* (prikaz ili naglašavanje određenih elemenata interfejsa) i *navigation* (promena putanje učenika kroz kurs).

Pravila kojima se donose odluke o daljoj personalizaciji su:

$Personalisation(?p) \wedge basedOn(?p, ?c) \wedge Condition(act) \wedge CurrentGoal(?g)$  PA4  
 $isTypeOf(?g, stylematch) \rightarrow determines(?p, act100)$

$Personalisation(?p) \wedge basedOn(?p, ?c) \wedge Condition(ref) \wedge CurrentGoal(?g)$  PA5  
 $isTypeOf(?g, stylematch) \rightarrow determines(?p, ref100)$

$Personalisation(?p) \wedge basedOn(?p, ?c) \wedge Condition(act) \wedge CurrentGoal(?g)$  PA6  
 $isTypeOf(?g, adaptinterface) \rightarrow determines(?p, act101)$

$Personalisation(?p) \wedge basedOn(?p, ?c) \wedge Condition(ref) \wedge CurrentGoal(?g)$  PA7  
 $isTypeOf(?g, adaptinterface) \rightarrow determines(?p, ref101)$

$Personalisation(?p) \wedge basedOn(?p, ?c) \wedge Condition(act) \wedge CurrentGoal(?g)$  PA8  
 $isTypeOf(?g, navigation) \rightarrow determines(?p, act102)$

$Personalisation(?p) \wedge basedOn(?p, ?c) \wedge Condition(ref) \wedge CurrentGoal(?g)$  PA9  
 $isTypeOf(?g, navigation) \rightarrow determines(?p, ref102)$

Na osnovu trenutnog stila učenja i aktuelnog tipa adaptacije, sistem donosi odluku koje su instance klase *Decision* aktivne. Značenja pravila su: odluka pod nazivom *act100* se aktivira kada je trenutni učenik sa aktivnim stilom učenja i kada se primenjuje *styleMatch* tip adaptacije (pravilo adaptacije PA4), odluka pod nazivom *ref101* kada je trenutni učenik sa refleksivnim stilom učenja i kada se primenjuje *adaptInterface* tip adaptacije (pravilo adaptacije PA5), itd.

Dalja personalizacija zavisi od odluke koja se beleži u klasi *Decision* i na osnovu prethodnih pravila.

### **Pravila adaptacije u okviru domena obrade informacija: Aktivni i Refleksivni učenici**

U okviru domena *Procesiranja informacija* (eng. *Information processing*) razlikuju se dve kategorije učenika: *reflektori* (orijentisani na primere) i *aktivisti* (orijentisani na aktivnosti) (Kolb, 1984).

Aktivisti najlakše razumevaju gradivo tako što obavljaju različite aktivnosti vezane za njega – raspravljaju, komentarišu ili objašnjavaju drugima.

Reflektori su osobe koje imaju naviku da sakupljaju i analiziraju podatke pre nego što preduzmu konkretnu akciju. Oni radije pregledaju mišljenja drugih ljudi ili stručnjaka nego što sami obavljaju pojedine aktivnosti.

U sistemu učenicima sa aktivnim stilom učenja se prikazuju prvo aktivnosti, zatim teorija, objašnjenje i onda tek na kraju primeri (Klašnja-Milićević et al., 2011b).

Primer pravila koja prilagođavaju redosled resursa na osnovu aktivnog stila učenja su:

$Resource(?x) \wedge isTypeOf(?x, exercise) \wedge Resource(?y) \wedge isTypeOf(?y, example) \wedge Decision(act100) \rightarrow hasPrerequisite(?y, ?x)$  PA10

$Resource(?x) \wedge isTypeOf(?x, theory) \wedge Resource(?y) \wedge isTypeOf(?y, example) \wedge Decision(act100) \rightarrow hasPrerequisite(?x, ?y)$  PA11

Svrha prethodnih pravila je definisanje preduslova koji se moraju ispuniti prilikom inicijalizacije korisničke sesije. Značenje pravila adaptacije PA10 je: ako je za aktivnog učenika doneta odluka o adaptaciji pod oznakom *act100* tada resursi za prikaz vežbe (eng. *exercise*) imaju prioritet u odnosu na resurse za prikaz primera (eng. *example*). Stoga, pravilo adaptacije PA10 definiše da je savladavanje vežbe (*exercise*) preduslov za prelazak na predstavljanje primera (*example*).

Pravilo adaptacije PA11 definiše prioritet resursa za prikaz teorije (*theory*) nad prikazivanjem primera (*examples*).

Za učenika sa refleksivnim stilom učenja redosled prikaza resursa je drugačiji – prvo mu se prikazuje primer (*example*), zatim objašnjenje (*explanation*) pa teorija (*theory*) i tek na kraju se od njega zahteva da izvede aktivnost (*activity*).

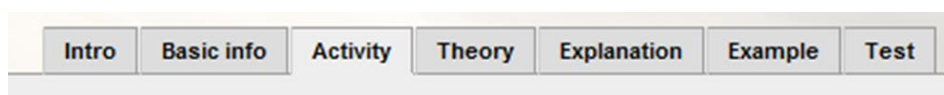
Primer pravila koja prilagođavaju redosled resursa na osnovu refleksivnog stila učenja su:

$Resource(?x) \wedge isTypeOf(?x, exercise) \wedge Resource(?y) \wedge isTypeOf(?y, example) \wedge Decision(ref100) \rightarrow hasPrerequisite(?x, ?y)$  PA12

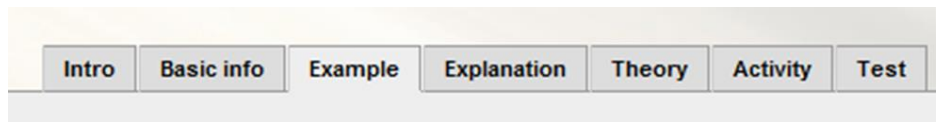
$Resource(?x) \wedge isTypeOf(?x, theory) \wedge Resource(?y) \wedge isTypeOf(?y, example) \wedge Decision(ref100) \rightarrow hasPrerequisite(?y, ?x)$  PA13

Značenje pravila PA12 i PA13 su analogni pravilima PA10 i PA11 ali za refleksivni stil učenja. Iz ugla učenika, rezultat pokretanja ovih pravila je prikaz nastavnih resursa u odgovarajućem redosledu.

Primeri korisničkog interfejsa za učenike sa aktivnim i refleksivnim stilom učenja su prikazani na slikama 32 i 33, respektivno.



Slika 32. Korisnički intefejs za učenike sa aktivnim stilom učenja



Slika 33. Korisnički interfejs za učenike sa refleksivnim stilom učenja

Drugi način personalizacije za domen *Procesiranja informacija* je naglašavanje linkova (eng. *link annotation*). U ovom pristupu se svakom resursu dodeljuje odgovarajuća uloga. Na primer, ako resursu dodelimo ulogu činjenica (*fact*) ili definicija (*definition*) onda se taj resurs koristi za povećavanje osnovnog znanja učenika (pretpostavka je da učenicima sa refleksivnim stilom učenja više odgovaraja konkretno iznošenje činjenica i definicija).

Sa druge strane, ako je uloga resursa primer (*example*) tada se taj resurs koristi za povećanje praktičnog znanja učenika (za aktiviste). Ako je odgovarajuća uloga resursa predodređena za učenike sa aktivnim stilom učenja, tada se taj resurs preporučuje učeniku. Odgovarajuće pravilo kojim se realizuje ta aktivnost je:

$$\text{Learner}(?x) \wedge \text{hasLearningStyle}(?x, \text{active}) \wedge \text{Resource}(?r) \wedge \text{ResourceRole}(?c) \wedge \text{supports}(?r, ?c) \wedge \text{supports}(?r, \text{active}) \rightarrow \text{isRecommended}(?r, \text{true}) \quad \text{PA14}$$

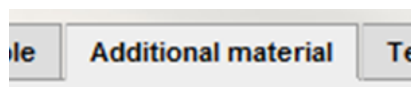
Značenje prethodnog pravila je: ako trenutni učenik ima aktivni stil učenja i ako resurs *r* ima definisanu ulogu za prikaz nastavnog materijala za aktivne učenike, tada se status tog resursa *r* za tog učenika postavlja na *preporučen* (eng. *isRecommended*). *isRecommended* je vrednosni atribut (eng. *datatype property*) klase *Resurs*.

Sa pravilom PA14 status resursa se menja i vrednost atributa *recommended* se postavlja na *true*. Na taj način se pokreće izmena korisničkog interfejsa.

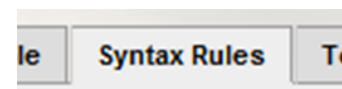
U ovom slučaju sistem vrši pretragu među resursima koji su predefinisani za učenike sa aktivnim stilom učenja. Na primer, učeniku sa aktivnim stilom učenja se preporučuju aktivnosti kao što su kvizovi, chat i forumi. Na osnovu toga sistem naglašava odgovarajuće linkove koji nude opcije komunikacije sa drugim učenicima i/ili predavačima (slika 34). Slična pravila se koriste prilikom adaptacije za ostale stilove učenja.



Slika 34. Preporuka opcije *Communication*



Slika 35. Preporuka opcije *Additional material*



Slika 36. Preporuka opcije *Syntax rules*

### Pravila adaptacije u okviru domena percepcije informacija: Osećajni i intuitivni učenici

U okviru domena percepcije informacija (eng. *Information perception*) postoje dve kategorije učenika: osećajni (eng. *sensing learners - sensors*) i intuitivni učenici (eng. *intuitive learners - intuitors*).

Učenici sa osećajnim stilom učenja imaju strpljenja sa detaljima i brzo memorišu činjenice. Sa druge strane, intuitivni učenici su bolji prilikom savladavanja novih koncepata i često su uspešniji sa apstraktnim primerima i matematičkim formulama.

Osećajni učenici vole da rešavaju probleme ustaljenim metodama i ne vole preterano komplikovanje i složena rešenja. Sa druge strane, intuitori vole inovacije i ne odgovara im

ponavljanje informacija. Osećajni učenici su praktičniji, pažljiviji i temeljniji dok intuitivni učenici rade brže i često su inovativniji i sa više ideja od osećajnih učenika.

Pretpostavlja se da će osećajni učenici biti zainteresovani za dodatni materijal, te im se pruža mogućnost prikaza takvog materijala preko korisničkog interfejsa (slika 35. Klašnja-Milićević et al., 2011b). Pravila koja pokreću ove akcije su:

$$\text{Learner}(\?x) \wedge \text{hasLearningStyle}(\?x, \text{sensing}) \wedge \text{Resource}(\?r) \wedge \text{isTypeOf}(\?r, \text{exercise}) \wedge \text{ConceptRole}(\?c) \wedge \text{supports}(\?r, \?c) \wedge \text{supports}(\?c, \text{sensing}) \rightarrow \text{isRecommended}(\?r, \text{true})$$
 PA15
$$\text{Learner}(\?x) \wedge \text{hasLearningStyle}(\?x, \text{sensing}) \wedge \text{Resource}(\?r) \wedge \text{isTypeOf}(\?r, \text{example}) \wedge \text{ConceptRole}(\?c) \wedge \text{supports}(\?r, \?c) \wedge \text{supports}(\?c, \text{sensing}) \rightarrow \text{isRecommended}(\?r, \text{true})$$
 PA16

Ova pravila adaptacije dodeljuju vrednost atributu koji određuje da li je resurs preporučen ili ne, na vrednost *true*. Pravila adaptacije PA15 i PA16 se koriste za generisanje preporuka za osećajne učenike. Ako je atribut resursa pod imenom *Recommended* postavljen na *true* tada se šalje informacija ontologiji nastavne strategije (eng. *Teaching strategy ontology*) sistema da je preporučen resurs odgovarajući za tog učenika te da se može njemu prikazati.

Učenicima sa intuitivnim stilom učenja se prikazuje apstraktni materijal i formule. Odgovarajuća objašnjenja se prikazuju u obliku blok dijagrama i preciznih sintaksnih pravila. Primer pravila koji implementira tu akciju je:

$$\text{Learner}(\?x) \wedge \text{hasLearningStyle}(\?x, \text{intuitive}) \wedge \text{Resource}(\?r) \wedge \text{isTypeOf}(\?r, \text{explanation}) \wedge \text{ConceptRole}(\?c) \wedge \text{supports}(\?r, \?c) \wedge \text{supports}(\?c, \text{intuitive}) \rightarrow \text{isRecommended}(\?r, \text{true})$$
 PA17

Pravilo adaptacije PA17 preporučuje resurs sa dodatnim objašnjenjima (eng. *explanation*) učeniku sa intuitivnim stilom učenja. Rezultat adaptacije je prikaz odgovarajuće kartice u panelu (eng. *tabbed pane*) (slika 36).

### Pravila adaptacije u okviru domena prijema informacija: Vizuelni i verbalni učenici

U okviru domena *Prijema informacija* (eng. *Information reception*) postoje dve kategorije učenika: vizuelni (eng. *visual*) i verbalni (eng. *verbal*) učenici.

Učenici sa vizuelnim stilom učenja najbolje zapamte ono što vide – slike, grafikone, dijagrame toka i druge vizuelne prikaze (Klašnja-Milićević et al., 2011b).

Učenici sa verbalnim stilom učenja više znanja steknu uz pomoć tekstualnih objašnjenja. Sledeća pravila generišu preporuke odgovarajućih resursa za oba ova stila učenja:

$$\text{Learner}(\?x) \wedge \text{hasLearningStyle}(\?x, \text{visual}) \wedge \text{Resource}(\?r) \wedge \text{ConceptRole}(\?c) \wedge \text{supports}(\?r, \?c) \wedge \text{supports}(\?c, \text{visual}) \rightarrow \text{isRecommended}(\?r, \text{true})$$
 PA18
$$\text{Learner}(\?x) \wedge \text{hasLearningStyle}(\?x, \text{verbal}) \wedge \text{Resource}(\?r) \wedge \text{ConceptRole}(\?c) \wedge \text{supports}(\?r, \?c) \wedge \text{supports}(\?c, \text{verbal}) \rightarrow \text{isRecommended}(\?r, \text{true})$$
 PA19

Pravilo adaptacije PA18 se koristi za preporuku resursa koji sadrže slike i dijagrame vizuelnim učenicima (slika 37) dok pravilo adaptacije PA19 preporučuje resurs sa



tekstualnim objašnjenjem verbalnim učenicima (slika 38). Navedeni primeri verbalnog i vizuelnog prikaza dati su za kurs programiranja u Javi koji je realizovan u sistemu Protus 2.0. Detalji spomenutog kursa će biti prikazani u 7. poglavlju.

**While loop**

Intro Basic info Theory Explanation Example Syntax Rules Test

**While loop example:**

WHILE loop example:

```
int num = 1;
while (num <= 10) {
    System.out.println("Value of num is: " + num);
    num++;
}
```

start

```
graph TD
    Start((start)) --> Define[Define num  
int num = 1;]
    Define --> Decision{num <= 10}
    Decision -- [yes] --> Print[Print num]
    Print --> Inc[Inc num  
num++;]
    Inc --> Decision
    Decision -- [no] --> End((end))
```

Slika 37. Primer lekcije za učenike sa vizuelnim stilom učenja

**The FOR Statement**

Intro Basic info Theory Explanation Example Syntax Rules Test

**For loop**

The for statement also has another form designed for iteration through Collections and arrays. This form is sometimes referred to as the enhanced for statement, and can be used to make your loops more compact and easy to read.

To demonstrate, consider the following array, which holds the numbers 1 through 10:

```
int[] numbers = {1,2,3,4,5,6,7,8,9,10};
```

The following program, EnhancedForDemo, uses the enhanced for to loop through the array:

```
Class ForDemo{public static void main(String[] args){
    for(int i=1; i<11; i++){
        System.out.println("Count is: " + i);
    }
}
```

Slika 38. Primer lekcije za učenike sa verbalnim stilom učenja

### Pravila u okviru domena razumevanja informacija: Sekvencijalni i globalni učenici

U okviru domena *Razumevanja informacija* (eng. *Information understanding*) postoje dve kategorije učenika: sekvencijalni (eng. *sequential*) i globalni (eng. *global*) učenici. Sekvencijalni učenici se trude da prate logičan i postupan put ka postizanju rešenja.

Sa druge strane, učenici sa globalnim stilom učenja su spretniji u rešavanju kompleksnijih problema ako prethodno steknu globalnu sliku o gradivu, mada često imaju problem da objasne kako su do rešenja stigli.

Sekvencijalni učenici više vole da kroz kurs prolaze postepeno, linearno kroz lekcije koje se nadovezuju jedna na drugu, dok učenici sa globalnim stilom učenja radije uče u velikim koracima, često preskačući lekcije grabeći ka onim kompleksnijim.

Na osnovu ovih karakteristika stilova učenja, sekvencijalni učenici se kroz kurs vode prethodno definisanim rasporedom dok se globalnim učenicima nudi mogućnost slobodnog izbora lekcija i preskakanja resursa (Klašnja-Milićević et al., 2011b). Za definisanje redosleda koncepata, koriste se naredna pravila za adaptaciju za sekvencijalni stil učenja (učenicima sa globalnim stilom učenja se definiše raspored prikazivanja resursa tako da pravila postoje samo u cilju adaptacije za sekvencijalni stil učenja):

Learner(?x) ∧ hasLearningStyle(?x, explanation) → hasPrerequisite(loopStatements, explanation) PA20

Learner(?x) ∧ hasLearningStyle(?x, sequential) → hasPrerequisite(executionControl, loopStatements) PA21

Learner(?x) ∧ hasLearningStyle(?x, sequential) → hasPrerequisite(classes, executionControl) PA22

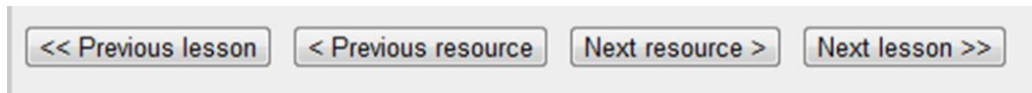
itd.

Pravila adaptacije PA20, PA21 i PA22 se koriste da učenicima sa sekvencijalnim stilovima učenja definišu raspored prikaza nastavnih resursa. Vrednostni atribut *hasPrerequisite*

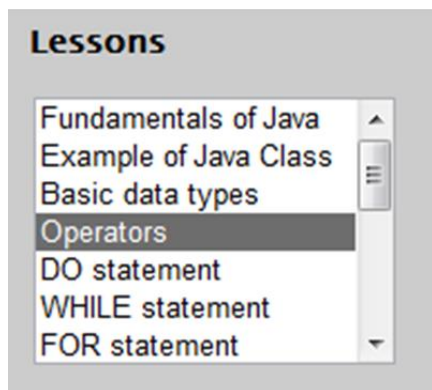
definiše redosled resursa. Na osnovu definisanih preduslova, sistem može da donosi odluke da li da predstavi lekciju u određenom redosledu (za učenike sa sekvencijalnim stilom učenja) ili da ponudi učeniku linkove ka svim lekcijama (učenicima sa globalnim stilom učenja).

U prvom slučaju se prikazuju elementi korisničkog interfejsa za sekvencijalnu navigaciju kroz kurs (u našem slučaju dugmad *Next/Previous resource/lesson*) što je i prikazano na slici 39.

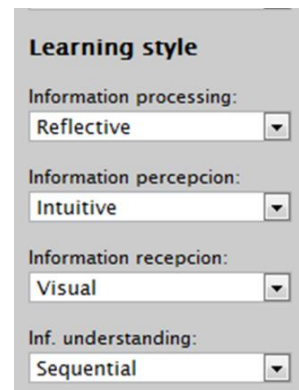
Sa druge strane, za učenika sa globalnim stilom učenja se prikazuju elementi korisničkog interfejsa za nesekvencijalnu navigaciju (slika 40).



Slika 39. Navigacija za učenike sa sekvencijalnim stilom učenja



Slika 40. Elementi za učenike sa globalnim stilom učenja



Slika 41. Opcije za promenu prikaza lekcije

U nastavku rada će biti prikazani primeri pravila adaptacije koja se koriste za formiranje modela učenika i ažuriranje ontologije modela učenika.

### 6.3.3 Pravila za formiranje modela učenika

U većini postojećih obrazovnih sistema, adaptacija prikaza nastavnog materijala na stilove učenika se bazira na inicijalnom stilu učenja aktivnog učenika. U daljem toku korišćenja sistema pretpostavlja se da će stil učenja aktivnog učenika ostati nepromenjen (Klašnja-Milićević et al., 2011b). Međutim istraživanja pokazuju da se stil učenja može menjati u zavisnosti od zadataka koje učenik savladava ili samog sadržaja nastavnog materijala. Stoga je kontraproduktivno ostaviti početno identifikovani stil učenja aktivnog učenika nepromenjen tokom celog kursa.

Sesija učenja se inicijalizuje na osnovu specifičnog stila učenja aktivnog učenika i generiše se sekvenca nastavnog materijala koji će mu biti prikazana. Nakon izbora lekcije koja će učeniku biti prikazana, sistem bira i tip resursa koji će biti prikazan na osnovu trenutnog stila učenja učenika. U daljem toku kursa, učeniku se ostavlja mogućnost da menja redosled pohađanja lekcija kao i mogućnost da izabere drugačiji tip prikaza svake lekcije (slika 41). Sistem nakon svake lekcije nudi učeniku mogućnosti provere znanja putem testa.

Sledeće pravilo ažurira ontologiju modela učenika na osnovu rezultata testa (Vesin et al. 2011b):

$$\text{Learner}(\?x) \wedge \text{Interaction}(\?y) \wedge \text{hasInteraction}(\?x, \?y) \wedge \text{Resource}(\?r) \wedge \text{resourceUsed}(\?y, \?r) \wedge \text{Performance}(\?p) \wedge \text{hasResult}(\?y, \?p) \wedge \text{hasGrade}(\?p, \?m) \wedge \text{swrlb:greaterThan}(\?m, 1) \rightarrow \text{isLearned}(\?r, \text{true}) \wedge \text{hasPerformance}(\?x, \?p) \quad \text{PA23}$$

Uz pomoć pravila PA23, sistem beleži rezultate interakcije učenika sa sistemom, dobijenu ocenu i korišćene resurse u okviru podataka o trenutnoj korisničkoj sesiji.

Promenjive  $x$ ,  $y$ ,  $r$ ,  $m$  i  $p$  predstavljaju instance klase učenika (*Learner*), interakcije (*Interaction*), resursa (*Resource*), ocene (*Grade*) i učinka (*Performance*), respektivno.

Resurs predstavlja prikaz dela lekcije kojem je učenik pristupio u trenutnoj sesiji.

Značenje pravila PA23 je: ako u bilo kom momentu upotrebe sistema, postoji učenik koji je posetio određeni resurs i tokom te interakcije proverio znanje na odgovarajućem testu, tada sistem beleži učinak tog učenika u odgovarajuću ontologiju i obeležava taj resurs kao savladan.

Pravilo PA23 se pokreće samo u slučaju da je učenik dobio pozitivnu ocenu.

U slučaju kada učenik ne prikaže potrebno znanje, pokreće se naredno pravilo:

$$\text{Learner}(\?x) \wedge \text{Interaction}(\?y) \wedge \text{hasInteraction}(\?x, \?y) \wedge \text{Resource}(\?r) \wedge \text{resourceUsed}(\?y, \?r) \wedge \text{Performance}(\?p) \wedge \text{hasResult}(\?y, \?p) \wedge \text{hasGrade}(\?p, \?m) \wedge \text{swrlb:equal}(\?m, 1) \rightarrow \text{hasExecuted}(\?x, \?r) \wedge \text{hasPerformance}(\?x, \?p) \quad \text{PA24}$$

Ovo pravilo obeležava resurs kao posećen ali ne menja njegov status u savladan, što znači da bi trebalo prikazati novi resurs koji podržava istu lekciju u nastavku korisničke sesije aktivnog učenika.

Ukoliko učenik ne prikaže potreban nivo znanja tokom sesije uz pomoć resursa koji podržavaju određeni stil učenja, njegov/njen inicijalni stil učenja će biti modifikovan uz pomoć narednog pravila:

$$\text{Learner}(\?x) \wedge \text{hasLearningStyle}(\?x, \text{verbal}) \wedge \text{Interaction}(\?i) \wedge \text{hasInteraction}(\?x, \?i) \wedge \text{Resource}(\?r) \wedge \text{resourceUsed}(\?i, \?r) \wedge \text{ResourceRole}(\?s) \wedge \text{hasRole}(\?r, \?s) \wedge \text{supports}(\?s, \text{verbal}) \wedge \text{Performance}(\?p) \wedge \text{hasResult}(\?i, \?p) \wedge \text{hasGrade}(\?p, \text{grade}) \wedge \text{swrlb:lessThan}(\text{grade}, \text{required}) \rightarrow \text{hasLearningStyle}(\?x, \text{visual}) \quad \text{PA25}$$

Promenjive  $x$ ,  $i$ ,  $r$ ,  $s$  i  $p$  predstavljaju instance klase učenika (*Learner*), interakcije (*Interaction*), resursa (*Resource*), uloge resursa (*ResourceRole*) i učinka (*Performance*), respektivno. Značenje pravila je: ako u bilo kom momentu, postoji učenik sa verbalnim stilom učenja koji koristi sistem i tokom interakcije sa sistemom posećuje odgovarajući resurs ali ne prikazuje potrebno znanje (potreban nivo znanja se čuva u globalnoj promenljivoj *required*), tada se kao stil učenja tog učenika menja u suprotan stil u okviru iste kategorije stilova.

U slučaju da je stil učenja trenutnog učenika vizuelan, tada će biti pokrenuto sledeće pravilo:

$$\text{Learner}(\?x) \wedge \text{hasLearningStyle}(\?x, \text{visual}) \wedge \text{Interaction}(\?i) \wedge \text{hasInteraction}(\?x, \?i) \wedge \text{Resource}(\?r) \wedge \text{resourceUsed}(\?i, \?r) \wedge \quad \text{PA26}$$

$\text{ResourceRole(?s)} \wedge \text{hasRole(?r,?s)} \wedge \text{supports(?s, visual)} \wedge$   
 $\text{Performance(?p)} \wedge \text{hasResult(?i,?p)} \wedge \text{hasGrade(?p, grade)} \wedge$   
 $\text{swrlb:lessThan(grade, required)} \rightarrow \text{hasLearningStyle(?x,verbal)}$

Pravilo adaptacije PA26 se pokreće za učenike sa vizuelnim stilom učenja. Značenje pravila je: ako tokom interakcije učenika  $i$  sa resursom  $r$  koji podržava prikaz nastavnog materijala učenicima sa vizuelnim stilom učenja, učenik ne dobije ocenu veću od prethodno definisane minimalne (koju može menjati administrator sistema ili je moguće definisati da se automatski menja na osnovu napretka učenika), onda je potrebno promeniti stil učenja tog učenika na verbalni (suprotan stil u okviru iste kategorije stilova).

Slična pravila se pokreću za ostale kategorije stilova učenja (intuitivne/osećajne, globalne/sekvencijalne i aktivne/refleksivne).

Sva SWRL pravila se pokreću uz pomoć Jess rules engine-a. Pravila se automatski pokreću neposredno nakon što se svi uslovi definisani u njima ispune. Nakon pokretanja odgovarajućih pravila, ažurirani podaci se zapisuju u OWL ontologije i na taj način se ažurira baza znanja. Dok se ontologije koriste da bi se povećala upotrebljivost i operativnost informacija iz domena, pravila adaptacije se koriste za predstavljanje logike adaptacije na način da predavači mogu imati uvid i lakše razumeti i modifikovati metode koje se obavljaju tokom personalizacije sistema.

#### 6.3.4 Adaptacija na osnovu pravila za generisanje preporuka

U opštem modelu tutorskog sistema, učeniku se prikazuje nastavni materijal u različitom rasporedu u zavisnosti od navigacione sekvence koja se odredi na osnovu stila učenja učenika (Vesin et al. 2011b). U ovom poglavlju će biti definisana SWRL pravila kojima se prilagođava raspored prikaza resursa učeniku.

Prilagođavanje rasporeda prikazivanja resursa (eng. *Resource sequencing*) predstavlja popularnu tehniku adaptacije u inteligentnim tutorskim sistemima (Janssen et al., 2007). Osnovna ideja je generisanje personalizovanog kursa za svakog učenika uz dinamički izbor optimalnih akcija, prezentacija, primera, zadataka i problema u svakom trenutku pohađanja kursa.

Pravila adaptacije sistema opšteg modela tutorskog sistema koja generišu preporuke mogu se podeliti u dve kategorije:

- oflajn pravila,
- pravila za odabir resursa.

Primeri pravila adaptacije iz prethodno navedenih kategorija biće prikazani u nastavku poglavlja.

##### Oflajn pravila

*Oflajn pravila* (eng. *off-line rules*) koriste podatke iz *Ontologije modela učenika* da bi prepoznali osobine učenika, njihove stilove učenja i na osnovu tih podataka generisali ciljeve kursa za te učenike. Učenici su grupisani u klastere, tj. kategorije kojima pripadaju na osnovu individualnih stilova učenja u okviru četiri definisane dimenzije stilova učenja. Na osnovu inicijalnog upitnika, svakom učeniku se dodeljuje jedna kombinacija stilova učenja (po jedna kategorije iz svake od četiri dimenzije stilova učenja). Na taj način

moгуće je formirati ukupno 2<sup>4</sup>, tj. 16 klastera (tabela 5). Klasteri se koriste za identifikaciju zajedničkih osobina i aktivnosti učenika koji pripadaju istom klasteru (Vesin et al. 2011a).

**Tabela 5. Identifikacija klastera na osnovu stilova učenja**

Broj klastera	Klaster 1	Klaster 2	Klaster 3	Klaster 4	.....	Klaster 16
Obrada informacija	Aktivni	Aktivni	Aktivni	Aktivni		Refleksivni
Percepcija informacija	Osećajni	Osećajni	Osećajni	Osećajni		Intuitivni
Prijem informacija	Vizuelni	Vizuelni	Verbalni	Verbalni		Verbalni
Razumevanje informacija	Sekvencijalni	Globalni	Sekvencijalni	Globalni		Globalni
<b>Oznaka klastera</b>	<b>cl1</b>	<b>cl2</b>	<b>cl3</b>	<b>cl4</b>	<b>.....</b>	<b>cl16</b>

Na početku procesa učenja, sistem odredi klaster kojem učenik pripada kao rezultat pokretanja odgovarajućeg pravila adaptacije. Na osnovu pripadnosti klasteru, sistem generiše listu preporuka nastavnog materijala i akcija učenicima koji pripadaju istom klasteru.

Na primer, učenici iz klastera *cl1* pripadaju kategorijama: aktivnih, osećajnih, vizuelnih i sekvencijalnih učenika u okviru domena: obrade informacija, percepcije informacija, prijema informacija i razumevanja informacija, respektivno. Zatim, učenici iz klastera *cl2* pripadaju kategorijama: aktivnih, osećajnih, vizuelnih i globalnih učenika u okviru domena: obrade informacija, percepcije informacija, prijema informacija i razumevanja informacija, respektivno. Tabela 5 sadrži kompletan prikaz kojem klasteru pripadaju učenici sa odgovarajućim stilovima učenja.

Za svaki klaster postoji odgovarajuće pravilo. Primeri pravila koje se pokreću ako učenik pripada klasteru sa oznakom *cl1* (pravilo adaptacije PA27) i *cl2* (pravilo adaptacije PA28) su dati u nastavku (analogna pravila se pokreću kada učenik pripada ostalim klasterima):

```
Learner(?x) ^ Performance(?p) ^ hasPerformance(?x,?p) ^ Condition(?c) ^ PA27
generates(?p,?c) ^ BehaviourPattern(?b) ^ include(?p,?b) ^
isTypeOf(?b,?n) ^ NavigationSequence(?n) ^ consistsOf(?n,a1) ^
consistsOf(?n,b2) ^ consistsOf(?n,c1) ^ consistsOf(?n,d1) ^
consistsOf(?n,e4) ^ swrlb:greaterThen(grade, required) → belong(?x,?cl1)
```

```
Learner(?x) ^ Performance(?p) ^ hasPerformance(?x,?p) ^ Condition(?c) ^ PA28
generates(?p,?c) ^ BehaviourPattern(?b) ^ include(?p,?b) ^
isTypeOf(?b,?n) ^ NavigationSequence(?n) ^ consistsOf(?n,a1) ^
consistsOf(?n,b2) ^ consistsOf(?n, d1) ^ consistsOf(?n, e4) ^
consistsOf(?n, c1) ^ swrlb:greaterThen(grade, required) →
belong(?x,?cl1)
```

Pravila PA27 i PA28 određuju pripadnost učenika odgovarajućem klasteru, na osnovu posećenog navigacionog niza *n* koji sadrži resurse sa sledećim oznakama: {a1, b2, c1, d1, e4}, odnosno: {a1, b2, d1, e4, c1}. Navigacioni niz predstavlja listu resursa (resursi za prikaz uvodnih razmatranja, zadataka, objašnjenja, primera itd.) u određenom rasporedu. U ovom konkretnom primeru određeni su klasteri za oznakom *cl1* i *cl2*.

Promenjive *x*, *p*, *c*, *b* i *n* predstavljaju redom: učenika - *x* (klasa *Learner*), učinak - *p* (klasa *Performance*), uslov - *c* (klasa *Condition*), šablon ponašanja - *b* (klasa *BehaviorPattern*), i navigacioni niz - *n* (klasa *NavigationSequence*).

Klasa *Condition* sakuplja podatke o učinku učenika, njegovom stilu učenja i generiše odgovarajući tip personalizacije koji će biti primenjen. Generisana personalizacija, u stvari, predstavlja poseban navigacioni niz koji će biti predstavljen učeniku.

Značenje pravila PA27 je: ako u bilo kom momentu rada sistema, postoji učenik koji je imao interakciju sa sistemom pod određenim uslovima i tokom te interakcije je uspešno savladao određeni navigacioni niz, tada se taj niz može smatrati uspešnim za datog učenika te se učenik može svrstati u odgovarajući klaster. Određivanje klastera se realizuje jedino ako je učenik uspešno savladao navigacioni niz, tj. dobio pozitivnu ocenu.

Sistem tokom izvođenja kursa ažurira bazu podataka interakcija učenika sa sistemom. Baza podataka interakcije učenika sadrži podatke za izgradnju navigacionih šablona (eng. *sequential patterns*).

Naredno pravilo se koristi u cilju ažuriranja navigacionog niza:

$Learner(?x) \wedge Concept(?c) \wedge Resource(?r) \wedge hasResource(?c, ?r) \wedge$  PA29  
 $isLearned(?c, true) \wedge hasPerformance(?x, ?p) \wedge BehaviourPattern(?b) \wedge$   
 $NavigationSequence(?n) \wedge isTypeOf(?b, ?n) \rightarrow swrl:add(?n, ?r, ?p)$

Pravilo adaptacije PA29 dodaje posećeni resurs u navigacioni niz u trenutnoj sesiji.

SWRL funkcija  $swrl:add(?x, ?y, ?z)$  dodaje resurs  $y$  i detalje o učinku  $z$  u navigacioni niz  $x$ .

Značenje pravila PA29 je: ako u bilo kom momentu rada sistema, postoji učenik koji je uspešno savladao neki koncept koji sadrži određeni resurs, tada sistem treba da doda taj resurs i detalje o učinku učenika uspešnom navigacionom nizu.

### Pravila za odabir resursa

Pravila adaptacije proizvode listu resursa koji se preporučuju učeniku. Iz liste nastavnog materijala se generiše lista preporučenih akcija i resursa u cilju generisanja optimalne putanje koju treba da prođe učenik u savladavanju nastavnog materijala.

Modul za generisanje preporuka generiše listu preporučenih resursa na osnovu rangiranja tih nizova koje generiše sistem (Vesin et al. 2012b). Nizovi se rangiraju na osnovu ocena koje su učenici dobili na testovima nakon posete resursima u odgovarajućem redosledu.

Na primer, neka su resursi obeleženi oznakama  $a1, a2, \dots, b1, b2, \dots$  itd. Ako je utvrđeno da učenik pripada klasteru  $c1$ , to znači da će mu resursi biti predstavljeni u sledećem nizu  $\{a1, b2, c1, d1, e4\}$ . Analogni nizovi resursa se definišu za ostale klasterne.

Na osnovu inicijalnog niza resursa, sistem pronalazi resurs sa najvećim rejtingom (koji najviše doprinosi napretku učenika) te aktuelni niz treba da se dopuni sa tim resursom. Sistem produžava niz resursom sa najvećim rejtingom (detalji procesa su prikazani u poglavlju 7.4), te se postojećem nizu  $\{a1, b2, c1, d1, e4\}$  dodaje resurs sa oznakom  $f2$  i zatim se preporučuje učeniku u nastavku kursa.

Preporuka se generiše uz pomoć sledećeg pravila adaptacije:

$Learner(?x) \wedge Performance(?p) \wedge hasPerformance(?x, ?p) \wedge Condition(?c) \wedge$  PA30  
 $generates(?p, ?c) \wedge BehaviourPattern(?b) \wedge include(?p, ?b) \wedge$   
 $isTypeOf(?b, ?n) \wedge NavigationSequence(?n) \wedge belong(?x, ?c11) \rightarrow$

isRecommended(f2, true)

Značenje pravila PA30 je: ako u bilo kom momentu rada sistema, postoji učenik kojem je odgovarajući navigacioni niz resursa određen kao optimalan, tada bi sistem trebalo da preporuči učeniku resurs koji je naredni u tom nizu. Time se status resursa  $f2$  menja i resurs se označava kao preporučeni, te se link ka tom resursu prikaže ili naglasi. Praćenjem odgovarajućeg navigacionog niza resursa, učenik može nesvesno da utiče na izvršavanje pravila adaptacije u daljem toku rada sistema.

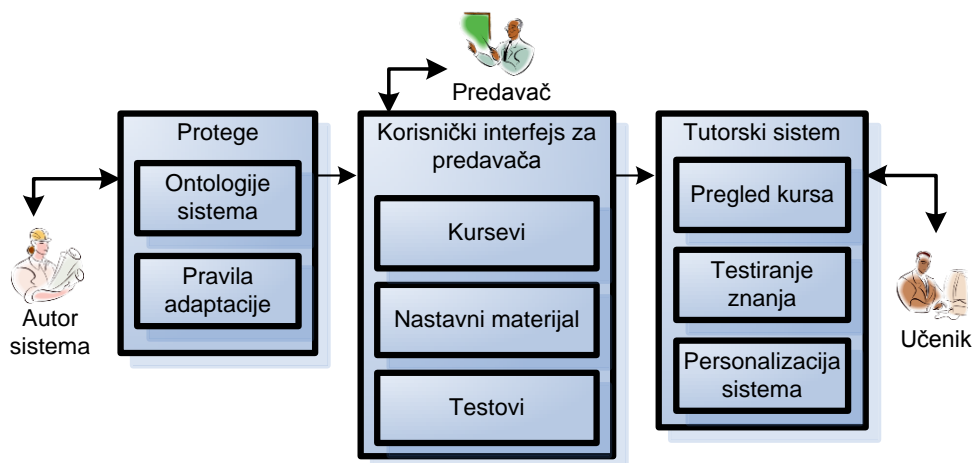
Ocenjivanje navigacionih nizova se vrši na osnovu ocena koje učenik dobije na kraju sesije za taj navigacioni niz. Tako se sistem prilagođava i prikazuje učenicima niz resursa koji su uticali da prethodni korisnici sistema sa istim karakteristikama i načinom učenja postignu bolje rezultate.

#### 6.4 Razvoj kurseva iz različitih domena

Model tutorskog sistema prikazan u disertaciji sastoji se od jasno definisanih i odvojenih komponenti koje se po potrebi mogu prilagođavati i dopunjavati. Moguća je i laka izmena opcija i mogućnosti personalizacije nastavnog materijala koji se nudi učenicima uz pomoć eksplicitno definisanih i dopunjivih pravila adaptacije. Formalno definisane ontologije će omogućiti ponovnu upotrebu komponenti tutorskog sistema prilikom izgradnje sličnih sistema.

Arhitektura opšteg modela tutorskog sistema bazirana na definisanom modelu omogućuje razvoj kurseva iz različitih domena kroz tri faze (slika 42):

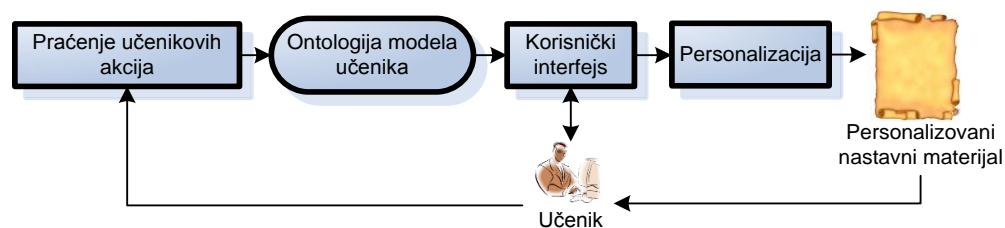
- kreiranje skeleta aplikacije u vidu skupa obrazovnih ontologija i pravila adaptacije,
- kreiranje pojedinačnih kurseva, odgovarajućeg nastavnog materijala za svaki pojedinačni kurs kao i skupa odgovarajućih testova za proveru znanja učenika,
- prezentacija personalizovanog nastavnog materijala prema potrebama svakog pojedinačnog učenika.



Slika 42. Razvoj kurseva iz različitih domena u opštem modelu tutorskog sistema

Tri faze razvoja kursa omogućuju jasno razdvajanje aktivnosti tri grupe projektanata sistema: autora personalizacije u tutorskom sistemu, autora nastavnog materijala i dizajnera korisničkog interfejsa sistema. Potpuno razdvajanje nastavnog sadržaja i njegovog dizajna sa jedne strane i aktivnosti dizajna aplikacije i prezentacije materijala sa

druge strane, omogućuje razvoj kurseva iz različitih domena i lakšu ponovnu upotrebu pojedinih komponenti aplikacije. Kursevi iz različitih domena odvijaju se u skladu sa sličnim postupkom i procesom učenja: od praćenja aktivnosti učenika, preko formiranja odgovarajućeg modela učenika, do personalizacije nastavnog materijala (slika 43).



**Slika 43. Aktivnosti učenika u opštem modelu tutorskog sistema**

Definisani opšti model tutorskog sistema omogućuje razvoj kurseva iz različitih domena podržanih različitim oblicima personalizacije korisničkog interfejsa i nastavnog materijala. Poglavlje 7 sadrži prikaz konkretizacije opšteg modela tutorskog sistema za formiranje kursa za učenje programskog jezika Java u okviru sistema Protus 2.0.



## Poglavlje VII

### DIZAJN I IMPLEMENTACIJA SISTEMA PROTUS 2.0

*Protus* je obrazovni softver iz kategorije tutorskih sistema, čija je osnovna namena prikaz personalizovanih kurseva iz različitih domena. Sistem je prvobitno zamišljen kao tutorski sistem opšte namene, međutim, prvo je u potpunosti implementiran kurs za učenje osnova programiranja u programskom jeziku Java (Vesin et al., 2009). Java je izabrana kao opšte prihvaćeni programski jezik na Prirodno-matematičkom fakultetu, Univeziteta u Novom Sadu i zato što predstavlja karakterističan primer objektno-orijentisanog jezika te je kao takav prikladan za učenje osnovnih koncepata objektno-orijentisanog programiranja. *Protus* je namenjen učenicima bez prethodnog znanja iz oblasti programiranja i nudi opcije pregleda nastavnog materijala i testiranja stečenog znanja.

Model tutorskog sistema definisan u poglavlju 6, omogućuje implementaciju neograničenog broja personalizovanih kurseva iz različitih domena kao i definisanje formalnih pravila adaptacije nastavnog materijala svakom pojedinačnom učeniku. Naredno poglavlje prikazuje konkretnu primenu opšteg modela za razvoj tutorskog sistema *Protus* 2.0 i realizacije kursa iz osnova programiranja.

#### 7.1 Prethodne verzije sistema *Protus*

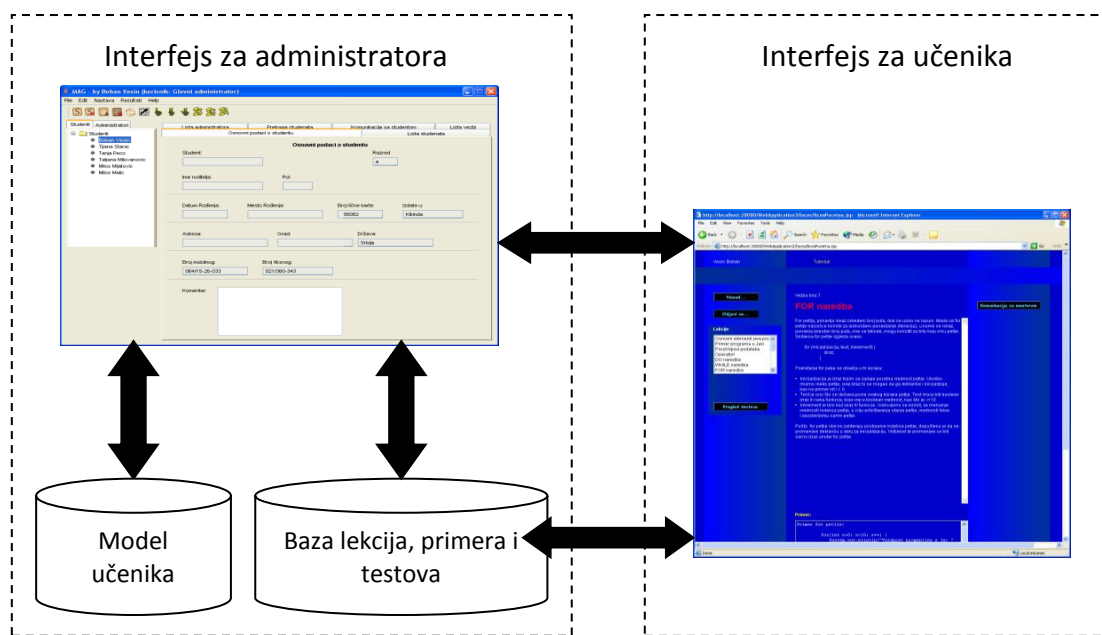
*Protus* 2.0 predstavlja unapređenje prethodnih verzija tutorskih sistema (tabela 6). Dva sistema su prethodila najnovijoj poboljšanoj i unapređenoj verziji i to:

- Tutorski sistem *Mag*. Ovaj sistem je bio zamišljen kao onlajn tutor za učenje sadržaja iz različitih domena. Sistem je korišćen za prezentaciju osnovnih koncepata programskog jezika *Java* kao i za proveru znanja učenika. Sistem *Mag* je sadržao samo osnovne oblike adaptacije i nije imao implementiran ni jedan od sistema za generisanje preporuka uključen u proces personalizacije (Vesin et al. 2009).

- Sistem Protus. Ovaj sistem predstavlja unapređenje prethodnog sistema. Osnovno unapređenje je postignuto realizacijom generisanja preporuka nastavnog materijala na osnovu identifikovanih stilova učenja za svakog pojedinačnog učenika.

Tabela 6. Funkcionalnosti tri verzije sistema

Funkcionalnosti sistema	Mag	Protus	Protus 2.0
Java onlajn kurs	X	X	X
Izveštaji o napretku učenika	X	X	X
Komunikacija učenika i mentora	X	X	X
Dodavanje novog nastavnog materijala	X	X	X
Mogućnost unosa dodatnih kurseva		X	X
Integrirani sistemi za generisanje preporuka		X	X
Identifikacija stilova učenja		X	X
Integrirane tehnologije semantičkog veba			X
Mogućnost ponovne upotrebe i deljenja nastavnog materijala			X
Mogućnost definisanja novih pravila adaptacije			X



Slika 44. Arhitektura sistema Mag

### 7.1.1 Sistem Mag

Mag predstavlja tutorski sistem za učenje sadržaja iz različitih domena (Vesin et al., 2009). Prvi implementirani kurs je bio za učenje osnova programiranja u programskom jeziku Java. Ideja je da se učenik koji pohađa kurs upozna sa osnovama Java programskog jezika, sintaksom, osnovnim elementima i naredbama. Takođe, cilj izrade sistema je bio da se celokupan kurs obavlja sa udaljenog računara uz pomoć Interneta i da se učeniku

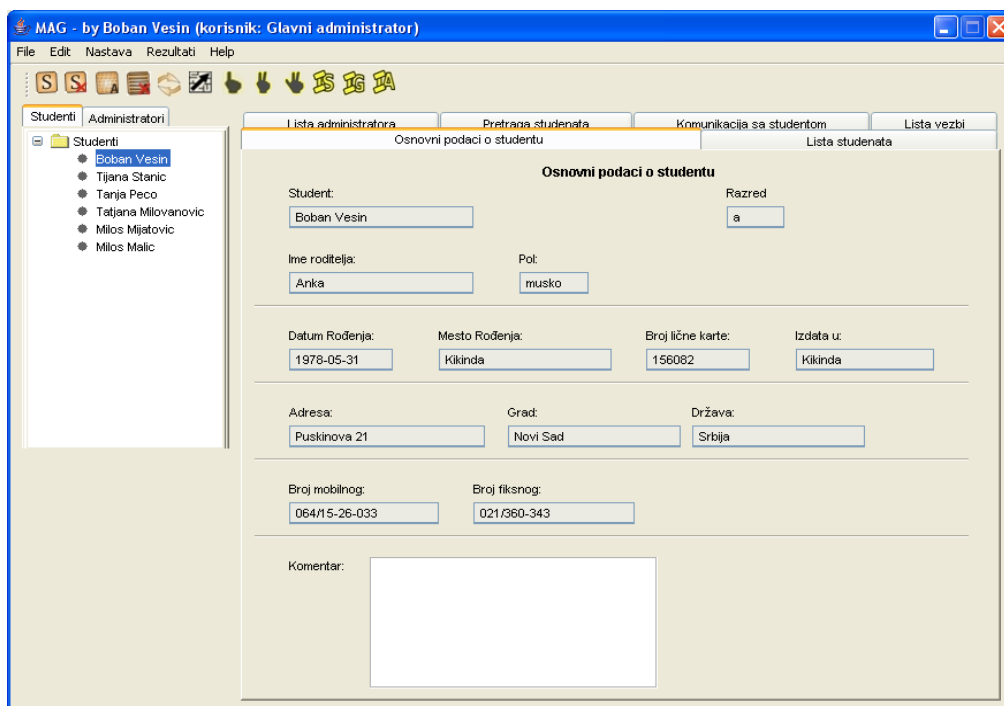
nakon svake lekcije ponudi mogućnost provere stečenog znanja. Sistem takođe nudi višestruke mogućnosti: pregled nastavnog materijala, testiranje znanja, onlajn programiranje, itd.

Sistem Mag se zasniva na centralizovanoj arhitekturi koja podrazumeva postojanje dva jasno odvojena dela sistema (slika 44):

- aplikacija na serverskoj strani sistema sa bazom lekcija, primera, testova i podacima o svakom pojedinačnom učeniku i
- niz veb stranica u formi interaktivnog kursa.

Na serverskoj strani sistema se nalazi aplikacija namenjena administratorima sistema (mentorima učenika) (slika 45). Osnovne funkcionalnosti ove aplikacije su:

- pregled i ažuriranje baze podataka učenika (dodavanje i brisanje podataka o učenicima polaznicima kursa),
- dobijanje izveštaja o napretku učenika (broju položenih i polaganih lekcija, uspehu na pojedinim lekcijama itd.),
- pregled i ažuriranje baze podataka administratora (mentora),
- dobijanje izveštaja o napretku učenika (učenici su podeljeni u razrede, svaki razred je dodeljen po jednom mentoru),
- laka pretraga podataka o učenicima po različitim kriterijumima,
- komunikacija sa učenicima (pregled dobijenih poruka i slanje novih),
- dodavanje novih oblasti i lekcija, kao i dodavanje novih primera i testova namenjenih pojedinim lekcijama.



Slika 45. Administrativni deo sistema Mag

Sistem koristi i ažurira bazu podataka sa podacima o učenicima, administratorima, testovima i lekcijama. Poruke koje razmenjuju učenik i mentor, kao i tekstovi tutorijala lekcija, primera i testova se čuvaju u odgovarajućem fajl sistemu.

Onlajn kurs se sastoji od niza *jsp* stranica namenjenih učeniku, koje omogućuju logovanje, prijavu novog učenika, pregled lekcija i primera, polaganje testova, pregled dotadašnjeg uspeha učenika kao i komunikaciju sa mentorom (slika 46).



Slika 46. Korisnički interfejs sistema Mag

### 7.1.2 Sistem Protus

Sistem Protus je nastao kao integracija sistema za generisanje preporuka u sistem Mag, sa ciljem efikasnije adaptacije nastavnog materijala potrebama svakog pojedinačnog učenika. Sistem nudi dve kategorije personalizacije, u obliku adaptivne hipermedije (eng. *adaptive hypermedia*) i sistema za generisanje preporuka (eng. *recommendation systems*). Dizajn i implementacija sistema za generisanje preporuka koji uzima u obzir karakteristike učenika i na osnovu njih generiše preporuke nastavnog materijala su predstavljeni u (Klašnja-Milićević et al., 2011b).

Modul sistema za generisanje preporuka je dizajniran da:

- prepoznaje pravilnosti u ponašanju učenika i identifikuje njihove stilove učenja,
- formira klastere (kategorije) učenika na osnovu njihovih stilova učenja i
- kategorizuje nastavni materijal na osnovu rejtinga i prezentuje preporučeni nastavni materijal učenicima.

Da bi sistemi za generisanje preporuka bili implementirani u novoj verziji sistema morali su biti zadovoljeni sledeći uslovi:

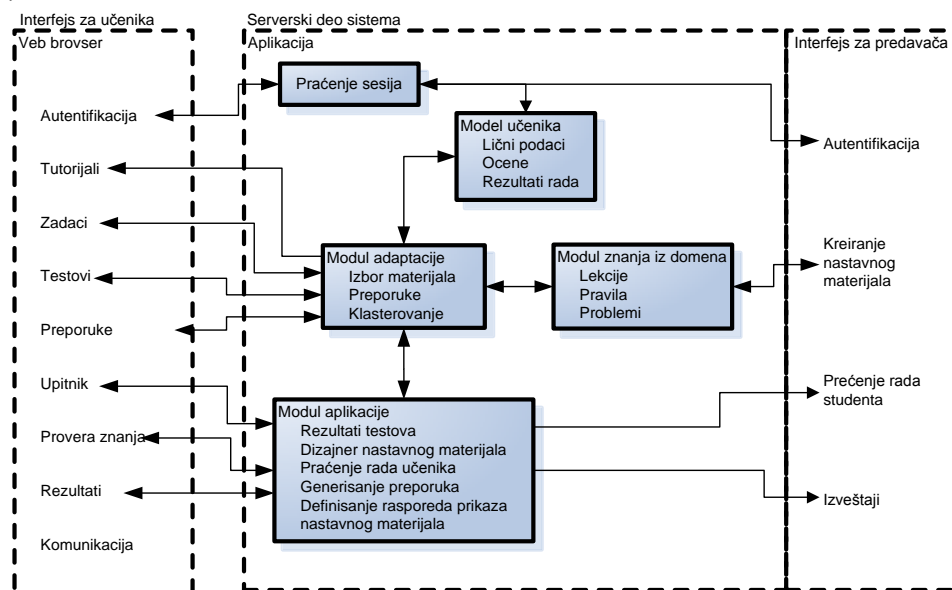
- postojanje dva odvojena korisnička interfejsa – za predavače i učenike,
- stroga razdvojenost osnovnih komponenti sistema: nastavnog materijala, podataka o učenicima, funkcija adaptacije i korisničkog interfejsa,
- neprekidno praćenje napretka učenika i razvijanje dinamičkog modela učenika,
- detaljno praćenje korisničkih sesija,
- mogućnosti komunikacije između predavača i učenika,
- procena znanja učenika i mogućnosti testiranja,
- funkcionalnosti za dodavanje novog nastavnog materijala,

- bogat semantički opis funkcionalnosti sistema u cilju bolje saradnje između komponenti sistema i
- mogućnosti efikasne koordinacije i komunikacije između komponenti sistema.

Arhitektura sistema Protus je bazirana na iskustvima autora drugih tutorskih sistema (Chen 2008, Merino & Kloos 2008, Šimić 2004) i arhitekture adaptivnih obrazovnih sistema prikazanih u (De Bra et al. 2004) i (Devedžić 2006). Slika 47 prikazuje opštu arhitekturu redizajniranog i proširenog sistema Protus. Arhitektura sistema je centralizovana i jezgro sistema čine modul adaptacije (eng. *adaptation module*), modul aplikacije (eng. *application module*) i modul znanja iz domena (eng. *domain module*) kao i model učenika (eng. *learner model*) koji se svi čuvaju na sistemskom serveru.

*Modul znanja iz domena* predstavlja skladište celokupnog nastavnog materijala, tutorijala i testova. Ovaj modul prikazuje način na koji su informacije strukturirane.

*Modul adaptacije* je odgovoran za izgradnju i ažuriranje karakteristika učenika koji se čuvaju u modelu učenika, kao i za personalizaciju nastavnog materijala koji se prezentuje učeniku. Ovaj modul procesira promene u karakteristikama učenika na osnovu njegovog ponašanja i sprovodi adaptaciju vizuelnih delova korisničkog interfejsa. Takođe, modul upravlja nastavnim materijalom, prezentuje materijal učeniku, generiše izveštaje o radu učenika, itd.



Slika 47. Arhitektura sistema Protus

*Model učenika* sadrži statičke i dinamičke podatke o svakom pojedinačnom učeniku. Statički podaci uključuju lične podatke o učeniku, ciljeve učenja, itd. Dinamički podaci su rezultati testiranja znanja, identifikovani stilovi učenja, vreme provedeno na pojedinim lekcijama i nastavnom materijalu, itd. Pored svega toga, model učenika sadrži podatke o sesijama učenja i istoriju pristupa nastavnom materijalu. Protus koristi te informacije u cilju predviđanja ponašanja učenika i adaptacije na njegove potrebe.

Sistem za praćenje sesija (eng. *session monitor*) konstantno prati akcije učenika tokom sesija i beleži ih u odgovarajući model učenika u cilju praćenja njegovih akcija i napretka. Na kraju svake sesije, podaci o učeniku i njegovim akcijama se beleže u model učenika. Neke od statičkih informacija i podataka o stilu učenja su dostupne učeniku i on ih može

proizvoljno menjati. Podaci iz modela učenika se koriste prilikom inicijalizacije naredne sesije.

Modul aplikacije (eng. *application module*) izvodi samu adaptaciju, tj. modul adaptacije prati instrukcije definisane u modulu aplikacije. Ove dve komponente su razdvojene u sistemu Protus da bi se olakšalo dodavanje novih pravila adaptacije. Na primer, u modulu aplikacije se donosi odluka o prikazu određenog nastavnog materijala učeniku a zatim modul adaptacije prikazuje materijal učeniku.

Sistem Protus nudi dve odvojene komponente, namenjene dvema kategorijama korisnika:

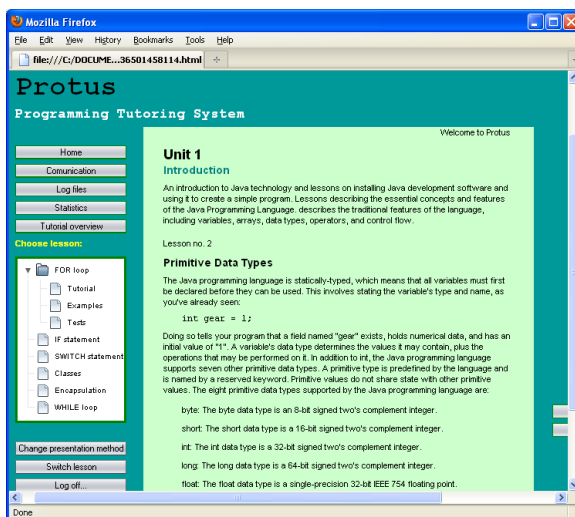
- učenici koji pohađaju osnovni programerski kurs iz jezika Java i
- predavači (autori kursa) koji predstavljaju administratore sistema i prate napredak učenika te im pomažu tokom procesa učenja.

Kao i u prethodnoj verziji sistema – u Magu, i u Protusu su definisana dva odvojena korisnička interfejsa za učenike i predavače (Vesin et al. 2009). Interfejs za predavača pomaže u procesu upravljanja nastavnim materijalom i podacima o učenicima. Ova komponenta sistema nije doživela velike promene u odnosu na prethodnu verziju. Unapređenja su izvršena samo u pogledu naprednije funkcije izveštavanja o napretku učenika.

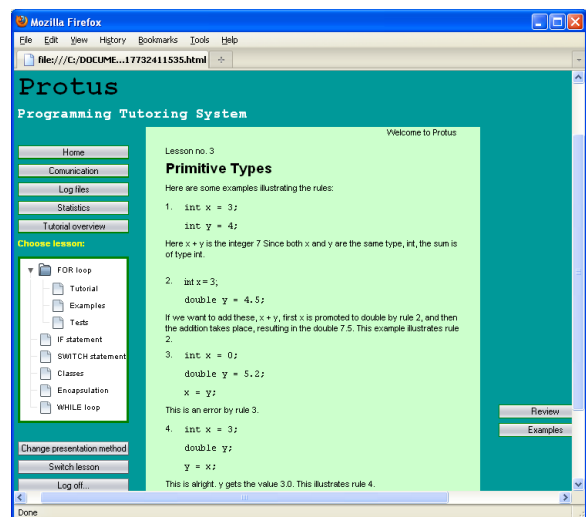
Interfejs za učenika predstavlja niz veb stranica za pregled nastavnog materijala i testiranje znanja. Svi podaci o učeniku, njegovom napretku kroz kurs kao i sam nastavni materijal se čuvaju na sistemskom serveru.

### Korisnički interfejs za učenika sistema Protus

Za svaku lekciju u sistemu, učenik prati niz određenih aktivnosti. Na početku lekcije, učeniku se prikazuju uvodni tekst (slika 48) i niz odgovarajućih primera (slika 49). Na kraju lekcije, zadaje se test učeniku i proverava stečeno znanje.



Slika 48. Primer tutorijala



Slika 49. Prikaz primera u Protusu

### Korisnički interfejs za predavača sistema Protus

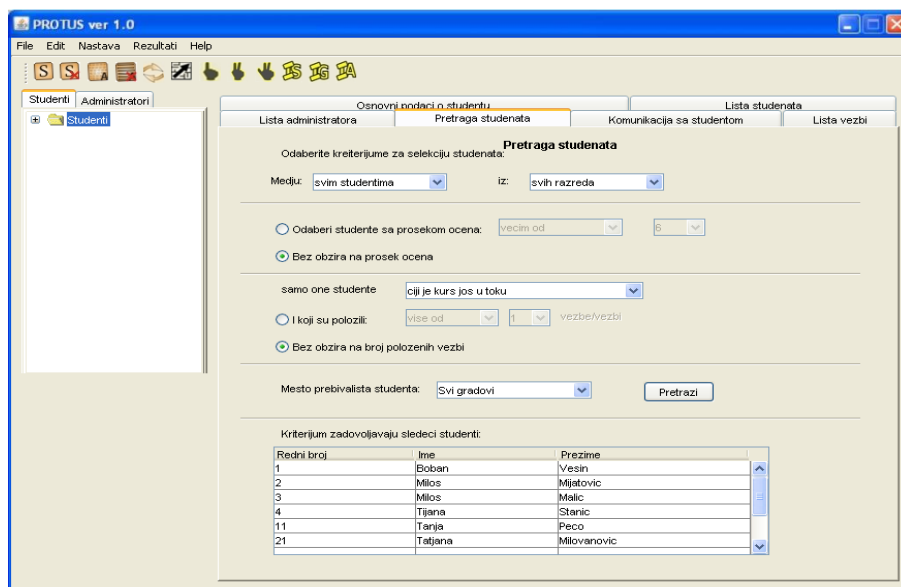
Protus nudi brojne korisne funkcionalnosti i predavačima tj. administratorima sistema kao i autorima nastavnog materijala. Protus se može koristiti za dobijanje izveštaja o aktivnostima učenika, uspehu na testovima i kretanjima kroz nastavni materijal. U

svakom slučaju, sistem nudi povratne informacije korisne za unapređenje kursa i samog nastavnog materijala.

Cilj Protusa je da pomogne predavačima i autorima kursa u proceni kvaliteta nastavnog materijala i samog dizajna kursa koji predaju. Sistem nudi predavačima brojne izveštaje o bitnim aspektima procesa učenja koji se sprovodi u sistemu. Izveštaji nastaju nakon analize podataka sakupljenih u modelu učenika i modulu za praćenje sesija. Tačnije, Protus nudi izveštaje o:

- aktivnostima učenika tokom korišćenja sistema,
- upotrebi pojedinačnih funkcija sistema i savladavanju nastavnog materijala predstavljenog u sistemu,
- specifičnosti interakcije učenika i sistema.

Na slici 50 je prikazan korisnički interfejs dela sistema namenjenog predavačima na kojem su prikazani statistički podaci o upotrebi sistema i ocenama učenika. Pomoću ove forme vrši se pretraga podataka i prikazuju odgovarajući rezultati predavaču. Predavač može da koristi parametre i filtere da bi dobio izveštaje koji se prikazuju u formi tabela i grafikona. Tip grafikona zavisi od izbora filtera. Na primer, predavač može da vidi koji material je u većoj meri korišćen, broj učenika sa pojedinačnim kategorijama stilova učenja ili pregled ocena koje su učenici dobili za pojedinačne lekcije. Izveštaji se mogu dobiti kako za pojedinačne učenike tako i za grupe učenika.



Slika 50. Interfejs za predavača sistema Protus

## 7.2 Razvoj programerskog kursa u sistemu Protus 2.0

Prilikom izgradnje komponenti sistema Mag i Protus korišćene su različite tehnologije. Na taj način komponente sistema nisu mogle biti ponovo upotrebljene i javljala se potreba za redefinisanjem koda bez obzira što su komponente obavljale praktično istu ulogu. Tako je, na primer, u Protusu morao biti promenjen i format nastavnog materijala u cilju optimizacije sistema, pa nije bilo moguće koristiti neki od već postojećih. Takođe, upotrebom različitih tehnologija se sprečava i dalje deljenje nastavnog materijala sa drugim sistemima. Da bi se sistem učinio šire dostupnim i da bi se omogućio njegov lakši razvoj i nadogradnja, javlja se potreba da se svaka komponenta sistema definiše u formi



ontologija. Svaka ontologija bi obavljala neku od funkcija prvobitnih komponenti sistema. Na taj način se povećava i nivo apstrakcije i bilo bi moguće lakše razumevanje uloge svake komponente i lakša operabilnost između komponenti arhitekture.

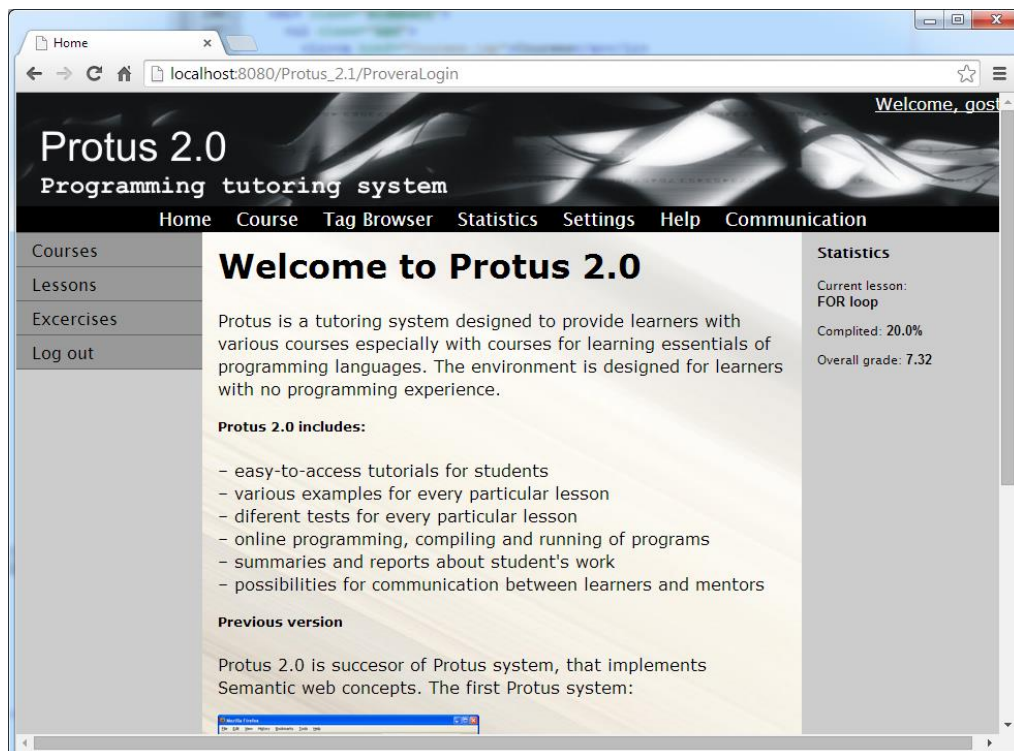
Opšti model tutorskog sistema prikazan u šestom poglavlju predstavlja osnovu razvoja nove verzije sistema Protus. Sistem Protus 2.0 omogućuje razvoj kurseva iz različitih domena. Svaki kurs se sastoji iz niza lekcija (konceptata) koje prikazuju pojedinačne segmente domena koji se obrađuje. Svaka lekcija se sastoji od niza resursa koji predstavljaju fajlove sa opisima pojedinačnih delova lekcija (uvodna razmatranja, objašnjenja, primeri, zadaci za vežbu, itd.). Za svaku lekciju se vezuje jedan ili više odgovarajućih testova za proveru znanja učenika. Na osnovu rezultata testova, utvrđuje se nivo napretka učenika, ažurira model učenika i generišu dalje opcije personalizacije.

Protus 2.0 se, kao i prethodne verzije ovog sistema, sastoji od dve osnovne komponente: korisničkog interfejsa za učenika i korisničkog interfejsa za predavača.

### 7.2.1 Korisnički interfejs za učenika

Učenik pohađa kurs preko veb aplikacije sistema Protus 2.0 (slika 51). Korisnički interfejs ovog dela sistema nudi učeniku sledeće funkcionalnosti:

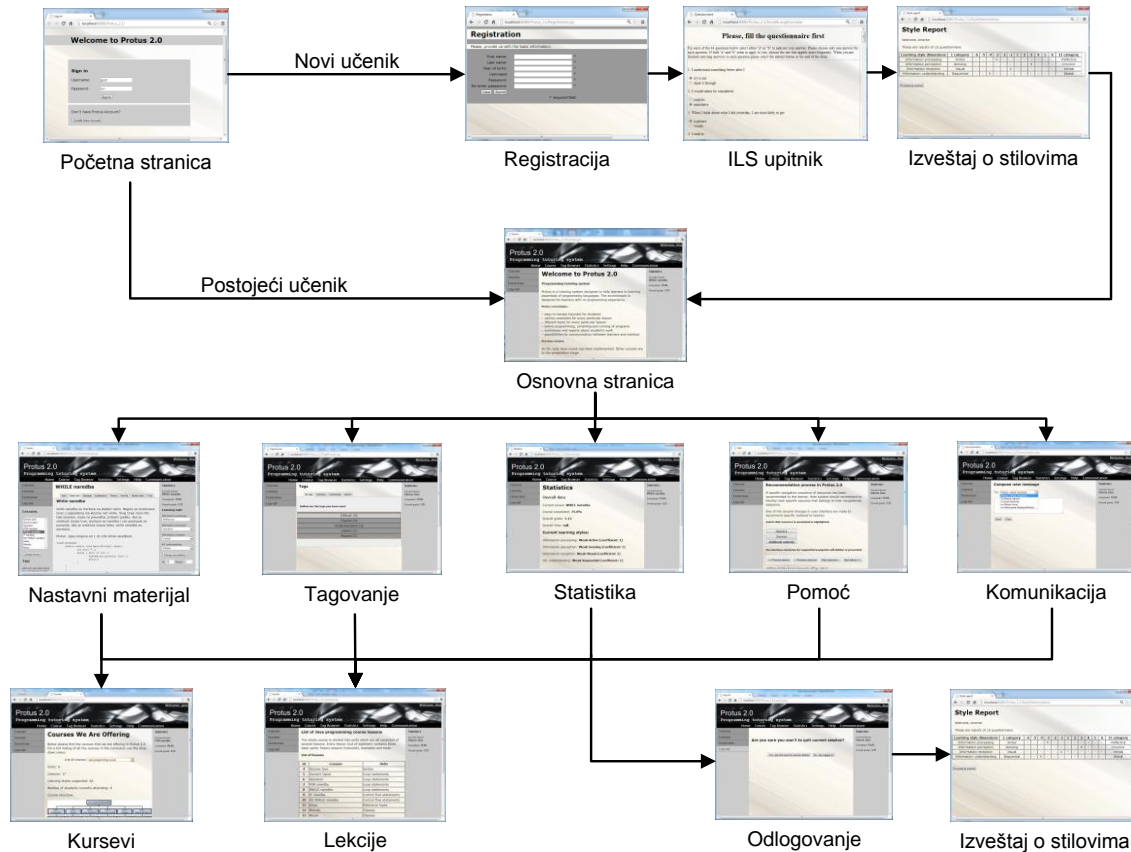
- pregled ponuđenih kurseva (ukoliko na sistemu postoje materijali za više kurseva) i nastavnog materijala,
- testiranje znanja,
- komunikaciju sa mentorom i drugim učenicima,
- promenu forme prikaza nastavnog materijala prilagođenog različitim stilovima učenja,
- izveštaje o napretku, rezultatima testova, toku kursa i promenama sopstvenih stilova učenja.



Slika 51. Korisnički interfejs sistema Protus 2.0



Korisnički interfejs za učenika u sistemu Protus 2.0 sadrži niz stranica koje nude osnovne funkcionalnosti učenicima. Raspored prikaza stranica se menja u zavisnosti od toga da li je učenik prvi put pristupio sistemu ili nastavlja već započeti kurs (slika 52). Detalji primenjene personalizacije nastavnog materijala u sistemu Protus 2.0 kao i detalji o upotrebi sistema će biti prikazani u poglavljima 7.4 i 7.5.

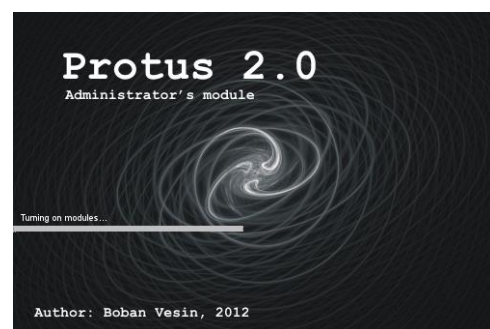


Slika 52. Mapa sajta sistema Protus 2.0

### 7.2.2 Korisnički interfejs za predavača

Administrator sistema koristi Protus 2.0 preko odgovarajuće desktop aplikacije (slika 53). Korisnički interfejs ovog dela sistema nudi administratoru sledeće funkcionalnosti:

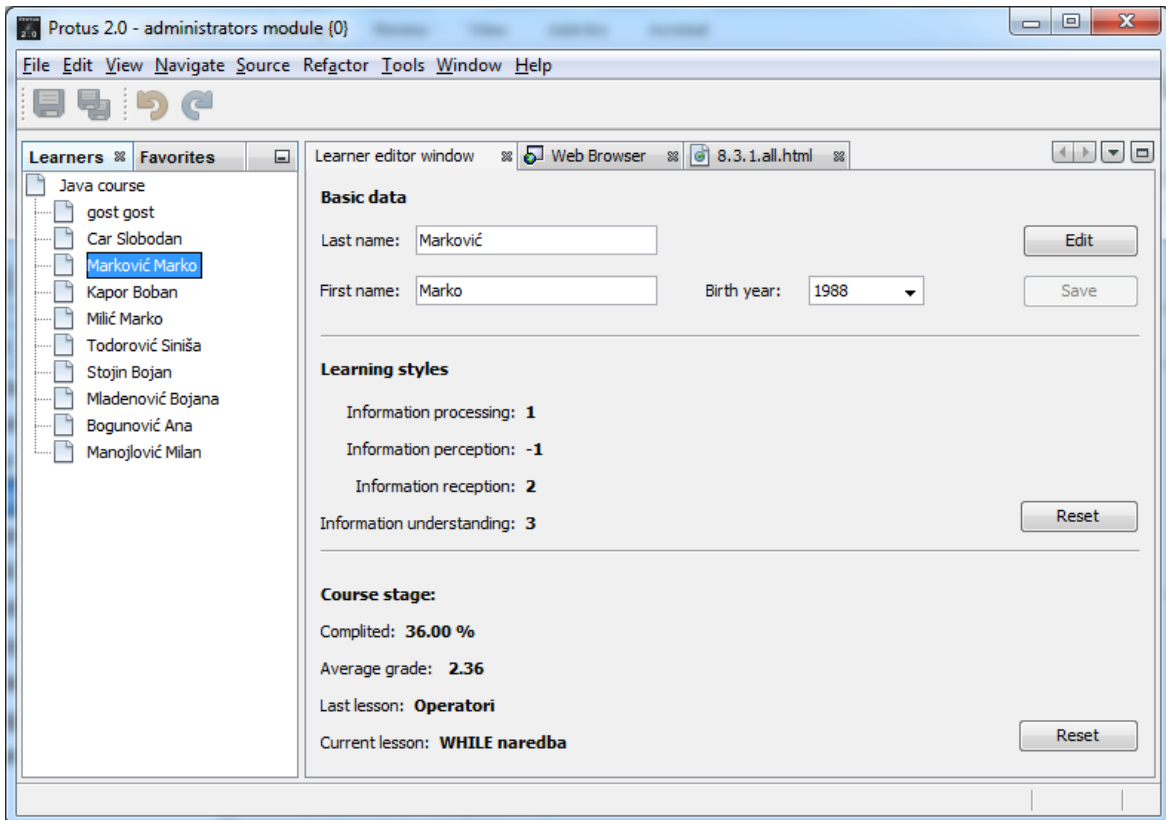
- pregled i izmenu osnovnih (statičkih) podataka o učenicima (slika 54),
- pregled postojećeg nastavnog materijala za postojeće kurseve (slika 55),
- unos novog nastavnog materijala za postojeće kurseve (slika 56),
- mogućnost dobijanja izveštaja o toku nastave, uspehu učenika, identifikovanim stilovima učenika, itd.



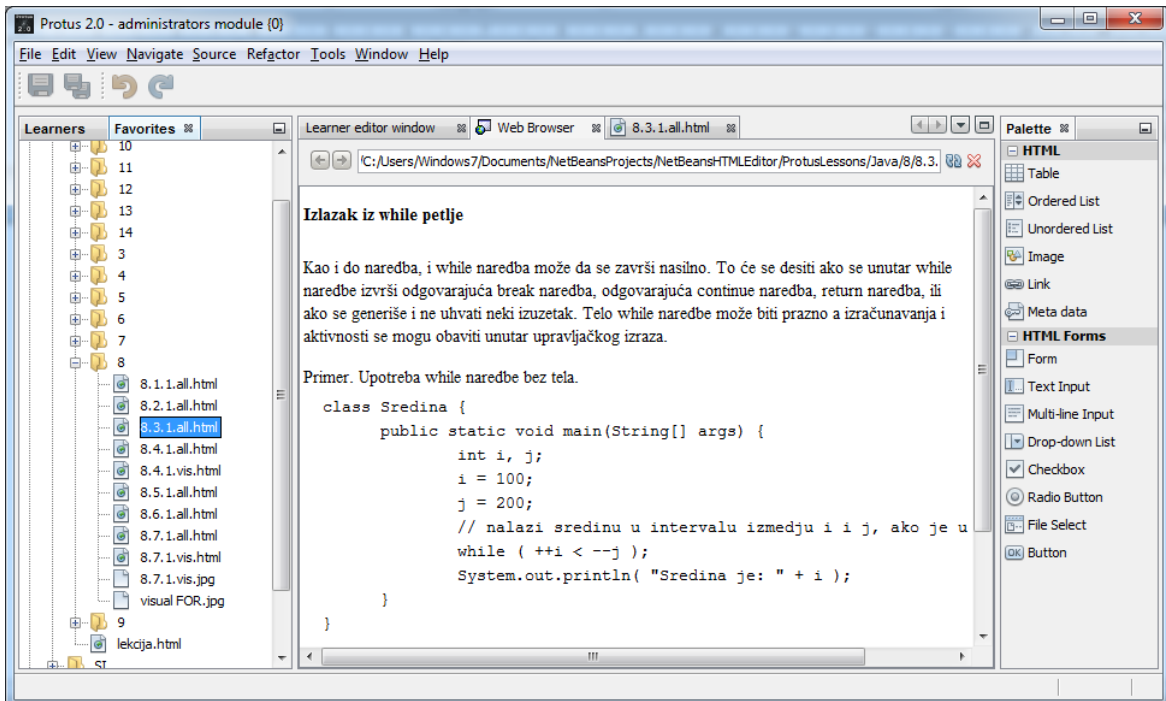
Slika 53. Uvodni prozor aplikacije

Sav nastavni materijal u sistemu Protus 2.0 je u formi *html* dokumenata. Sistem učitava dokument koji je određen u datom trenutku za aktivnog učenika na osnovu podataka iz

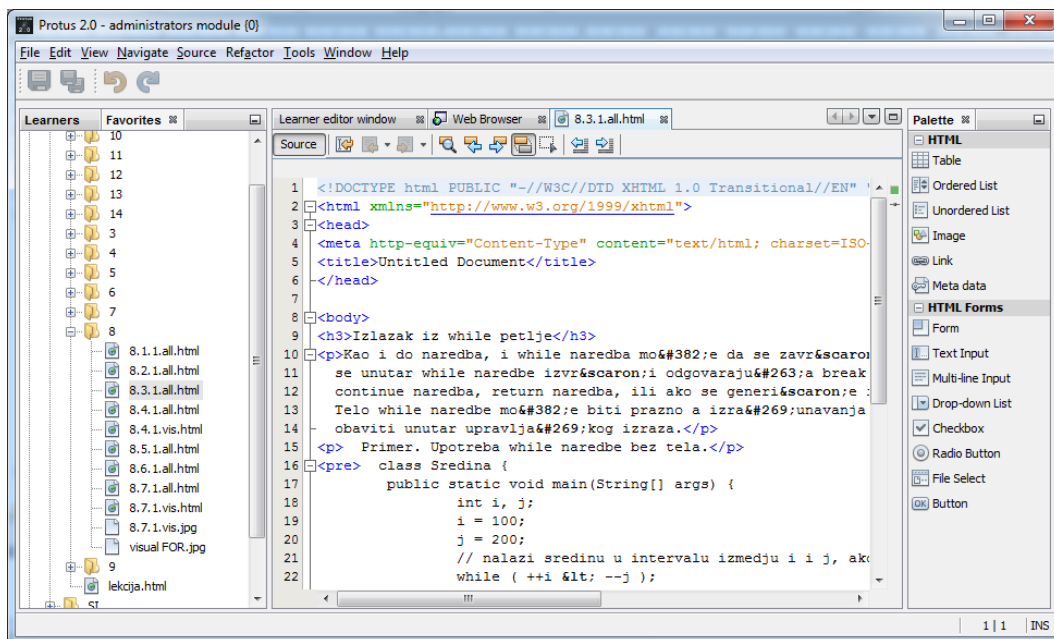
modela učenika i prikazuje ga u *web browser*-u. Administratorski deo sistema poseduje integrisani *html* editor koji omogućuje kreiranje i pregled nastavnog materijala (slika 56).



Slika 54. Osnovni prozor administrativnog dela sistema



Slika 55. Pregled postojećeg nastavnog materijala



Slika 56. Unos novog nastavnog materijala

### 7.3 Razvoj ontologija za programerski kurs iz Java programskog jezika

Osnovni cilj sistema za elektronsko učenje je da ostvari željenu pedagošku strategiju (Dehors & Faron-Zucker, 2006). Stoga, pedagoške ontologije se kombinuju sa upitnim jezicima i pravilima adaptacije da bi ponudile i podržale strategiju personalizacije. U sistemu Protus 2.0 strategija se sastoji od izbora nastavnog materijala koji će biti prikazan učeniku te se iz tih razloga mora definisati odgovarajuća formalna semantika ontologija.

Sistem Protus 2.0 je izgrađen na osnovu opšteg modela tutorskog sistema prikazanog u poglavlju 6. Model tutorskog sistema čine:

- ontologija za prikaz znanja iz domena – *ontologija znanja iz domena*,
- ontologija za izgradnju modela učenika – *ontologija modela učenika*,
- ontologija za prikaz aktivnosti u sistemu – *ontologija zadataka*,
- ontologija za prikaz nastavne strategije i pedagoških akcija – *ontologija nastavne strategije*,
- ontologija za definisanje tehnika za prilagođavanje korisničkog interfejsa – *ontologija korisničkog interfejsa*.

Ontologije u sistemu Protus 2.0 su pisane u jeziku OWL. U cilju razvoja ontologija i prevodenja u jezik OWL, korišćen je alat Protégé 4.3 (Protégé, 2012). Izvorni kod implementiranih ontologija generisan alatom Protégé 4.3 je prikazan u prilogu 2.

Da bi se razvio kompletan kurs iz osnova programskog jezika Java potrebno je kreirati odgovarajući nastavni materijal i integrisati ga u postojeći model tutorskog sistema.

U nastavku poglavlja je prikazan specifičan sadržaj ontologija za implementaciju Java programerskog kursa i aktivnosti sistema Protus 2.0 prilikom izvođenja kursa. Prikazani su detalji ontologije znanja iz domena, ontologije modela učenika i ontologije nastavne strategije, specifični za domen programiranja u Javi. Podaci koji se čuvaju u ontologiji zadataka i ontologiji korisničkog interfejsa kao i pravila adaptacije koja se nad njima izvršavaju ostaju nepromenjeni bez obzira na specifičnost domena.

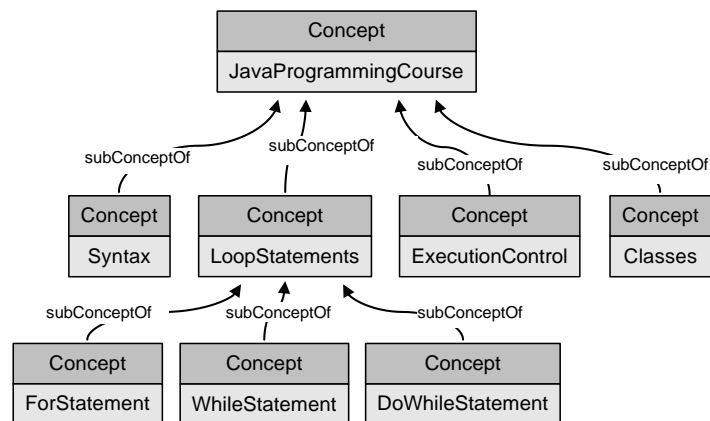
### 7.3.1 Ontologija znanja iz domena

Prvi kompletno implementiran kurs u sistemu Protus 2.0 je kurs za učenje programskog jezika Java. Nastavni materijal za ovaj kurs u sistemu Protus 2.0 sadrži 18 lekcija, tj. koncepata (eng. *concepts*) koji su grupisani u nastavne jedinice (eng. *Units*) kao što su: uvod, prikaz osnovne sintakse, naredbe grananja, naredbe ponavljanja, itd (slika 57).

Svakom konceptu je dodeljen veći broj resursa (eng. *resources*) u obliku tekstualnih fajlova, slika, ilustracija, grafikona, itd. Svaki resurs je definisan svojim tipom resursa (eng. *resource type*): teorija (eng. *theory*), primer (eng. *example*), zadatak (eng. *assignment*), vežba (eng. *exercise*), sintaksna pravila (eng. *syntax rules*), itd. Resursi se u sistemu Protus 2.0 čuvaju u obliku *html* dokumenata.

#### Koncepti (eng. *Concepts*)

Slika 57 prikazuje deo ontologije znanja iz konkretnog domena koja pokriva koncepte implementirane u kursu programskog jezika Java podržanog sistemom Protus 2.0 (Vesin et al. 2012b). Slika prikazuje korenski koncept sa nekim od svojih podkonceptata *Syntax*, *LoopStatements*, *ExecutionControl* i *Classes*. Koncept *LoopStatements* se dalje deli na koncepte: *ForStatement*, *WhileStatement* i *DoWhileStatement*. Jasna podela i kategorizacija koncepata i njihovih odnosa je neophodna za proces personalizacije.



Slika 57. Deo ontologije znanja iz domena u sistemu Protus 2.0

Unos lekcije u ontologiju domena se vrši kreiranjem nove instance klase *Concept*. Primer instance klase *Concept* koja sadrži informacije o lekciji *For Statement* je prikazana u tabeli 7.

Tabela 7. Primer instance klase *Concept*

Opis atributa	Naziv atributa	Vrednost atributa	Tip atributa
Identifikacioni broj koncepta	hasId	C009	Datatype property
Naziv koncepta	hasName	<i>For Statement</i>	Datatype property
Dodeljeni resursi	hasResource	R017	Object property
Nadklasa	subConceptOf	loopStatements	Object property
Preduslov	hasPrerequisite	ExecutionControl	Object property
Preduslov	hasPrerequisite	Syntax	Object property

Instanca klase *Concept* prikazana u tabeli 7. ima jedinstveni identifikator sa vrednošću C009, koristi se za definisanje lekcije pod nazivom *For Statement* i sadrži podatke o svojoj superklasi (ona je podkoncept koncepta *loopStatements*) i konceptima čije savladavanje je preduslov za prelazak na taj koncept (koncepti *ExecutionControl* i *Syntax*).

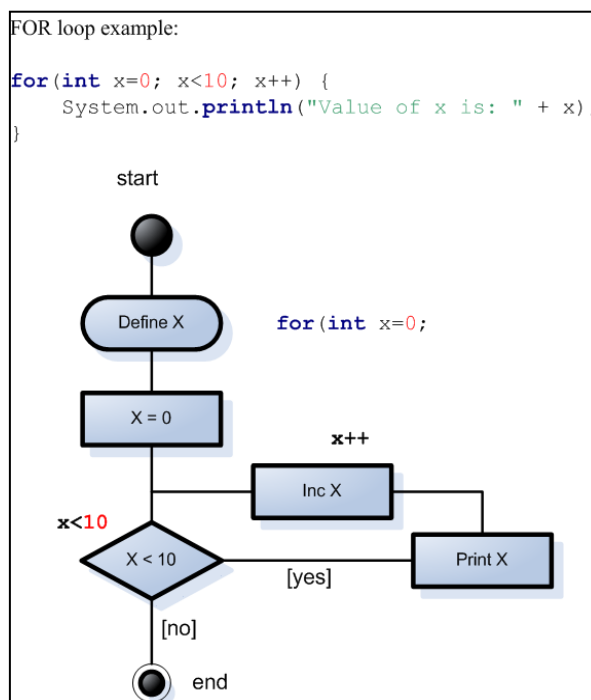
### Resursi (eng. Resources)

Podaci o konkretnim resursima se čuvaju u instancama klase *Resource*. Svaka instanca sadrži osnovne podatke o pojedinačnim resursima koji se koriste u procesu personalizacije. Takođe se definiše specifičan tip i uloga svakog resursa.

Primer instance klase *Resource* koja se koristi za prikaz sintakasnih pravila lekcije *For statement* data je u tabeli 8.

Tabela 8. Primer instance klase *Resource*

Opis atributa	Naziv atributa	Vrednost atributa	Tip atributa
Identifikacioni broj resursa	hasId	R017	Datatype property
Naziv resursa	hasName	forLoop017	Datatype property
Tip resursa	isTypeOf	Syntax rule	Object property
Tip koncepta	isResourceFor	For Statement	Object property
Uloga resursa	supports	Visual style	Datatype property
Da li je resurs posećen	isVisited	yes	Datatype property
Da li je resurs preporučen	isRecommended	no	Datatype property
Tip fajla	hasFileType	jpg	Datatype property
Uloga resursa	hasRole	definition	Datatype property
Link ka <i>jpg</i> fajlu	hasFigure	Figure6.jpg	Datatype property



Slika 58. Jpg resurs sistema Protus 2.0

Prikazana instanca klase *Resource* poseduje jedinstveni identifikator sa vrednošću R017. Instanca se koristi za prikaz sintaknih pravila prikazanih u lekciji (konceptu) pod imenom *ForStatement* i sadži link ka određenom *jpg* fajlu (slika 58) koji će biti prikazan učeniku ako sistem odabere ovaj resurs tokom procesa personalizacije.

### 7.3.2 Ontologija modela učenika

Tokom korisničkih sesija, realizuje se interakcija između učenika i sistema. Detalji o interakciji mogu pomoći prilikom zaključivanja o mogućim interesovanjima, ciljevima i znanju učenika. Ti se zaključci koriste prilikom personalizacije sistema. Stoga, ontologija mora da pruži mogućnosti praćenja i beleženja detalja o potencijalnoj interakciji učenika sa sistemom.

Sve akcije učenika u sistemu Protus 2.0 se beleže u klasi za prikaz interakcije (klasa *Interaction*). Interakcija podrazumeva sve relevantne akcije koje sprovodi učenik tokom određene sesije (klasa *Session*).

Primer instance klase *Interaction* koja se koristi za praćenje interakcije učenika tokom jedne sesije je prikazana u tabeli 9.

Tabela 9. Primer instance klase *Interaction*

Opis atributa	Naziv atributa	Vrednost atributa	Tip atributa
Identifikacioni broj interakcije	hasId	I007	Datatype property
Oznaka sesije	partOf	S1012	Object property
Tip interakcije	hasType	T02 (test)	Object property
Posećeni koncept	conceptUsed	C032	Object property
Učenik	whoInteracted	L01	Object property
Učinak	hasResult	P01	Object property

Instanca klase *Interaction* prikazana u tabeli 9 ima jedinstveni identifikator (atribut *hasId*) I007. Instanca klase je formirana tokom sesije sa oznakom S1012 kada je učenik sa jedinstvenim identifikatorom L01 polagao test i ostvario rezultate koji se čuvaju u instanci klase *Performance* sa oznakom P01. Instanca klase *Performance* sadrži, pored ostalih i podatke o ocenama koje je učenik dobio tokom testiranja. Na osnovu podataka iz klase *Performance*, sistem uz pomoć pravila adaptacije donosi odluke u daljem procesu personalizacije u okviru ontologije nastavne strategije (eng. *Teaching strategy ontology*) na način predstavljen u poglavlju 6.3.2 (pravila adaptacije: PA23 do PA26).

### 7.3.3 Ontologija nastavne strategije

Definisanje akcija adaptacije i personalizacije u tutorskom sistemu predstavlja kreiranje modela učenika i primenu različitih strategija i tehnika personalizacije u cilju efikasne adaptacije nastavnog materijala potrebama različitih učenika i njihovim individualnim stilovima učenja (Vesin et al. 2012b).

Primer instance klase *Condition* koja je formirana na osnovu podataka iz ontologije modela učenika je prikazana u tabeli 10. Prikazana instanca ima jedinstveni identifikator

I006 i sadrži podatke prikupljene na osnovu učenikovog stila učenja i prethodnih rezultata. Instanca se popunjava uz pomoć pravila adaptacije prikazanih u poglavlju 6.3.1 (pravila adaptacije: PA1 i PA2).

**Tabela 10. Primer instance klase *Condition***

Opis atributa	Naziv atributa	Vrednost atributa	Tip atributa
Identifikacioni broj uslova	hasId	I006	Datatype property
Stil učenja aktivnog učenika	generatedBy	LS03	Object property
Kategorija stila učenja	hasLearningStyleCategory	visual	Datatype property
Dimenzija stila učenja	hasLearningStyleDomain	Information Reception	Datatype property
Učinak aktivnog učenika	generatedBy	P01	Object property
Tip personalizacije	Generates	PR09	Object property

Sistem prati kretanje svakog učenika kroz nastavni materijal. Na osnovu posećenih resursa uz pomoć pravila adaptacije se ažurira instanca klase *BehaviourPattern* na način predstavljenim u poglavlju 6.3.3 (pravila adaptacije: PA29 i PA30). Instanca ove klase predstavlja specifičan tip klase *NavigationalSequence* koju čini niz resursa sa kojim je učenik imao interakciju.

Na primer, instanca klase *BehaviourPattern* sadrži informacije kakve su prikazane u tabeli 11. Ova konkretna instanca klase je tipa *NavigationalSequence* označena sa NS02, predstavlja niz resursa koje je posetio učenik tokom sesije S1012, sa rejtingom u vrednosti od 0,37 i na osnovu nje je generisana instanca klase *Personalization* označena sa PR09. Prisustvo ove konkretne instance klase *Personalisation* znači da će trenutno aktivnom učeniku biti prikazan nastavni materijal namenjen učenicima sa vizuelnim stilom učenja.

Sve akcije personalizacije u sistemu Protus 2.0 se primenjuju na osnovu prethodno prikazanih podataka iz klasa *Condition* i *BehaviourPattern* koje su vezane za konkretnog učenika. Primeri izvršene personalizacije uz pomoć SWRL pravila adaptacije na osnovu prikupljenih podataka su prikazani u poglavlju 6.3.3 (pravila adaptacije: PA27 i PA28).

**Tabela 11. Primer instance klase *BehaviourPattern***

Property description	Property name	Property value	Property type
Identifikacioni broj	hasId	BP0016	Datatype property
Navigaciona sekvenca	isTypeOf	NS02	Object property
Sesija	partOf	S1012	Datatype property
Personalizacija	generate	PR09	Object property
Rejting navigacione sekvence	hasRate	0,37	Datatype property

### 7.3.4 Ontologija zadataka i ontologija korisničkog interfejsa

*Ontologija zadataka* definiše koje su uloge pojedinih tipova koncepata i resursa dok *Ontologija korisničkog interfejsa* kreira niz resursa koji se preporučuju aktivnom učeniku i prikazuju na korisničkom interfejsu. Stoga, podaci koji se čuvaju u *Ontologiji zadataka* i *Ontologiji korisničkog interfejsa* kao i pravila adaptacije koja se nad njima izvršavaju ostaju nepromenjeni bez obzira na specifičnost domena.

## 7.4 Personalizacija učenja u sistemu

Različite tehnike se primenjuju da bi se nastavni sadržaji prilagodili učenicima na osnovu njihovog znanja, želja, stilova i ciljeva učenja. Protus 2.0 nudi dve osnovne kategorije personalizacije upotrebom sistema za generisanje preporuka (Ivanović et al. 2008):

- **Prilagođavanje sadržaja** (eng. *content adaptation*) – sadržaji se prikazuju na različite načine, uz pomoć različitih formi nastavnog materijala na osnovu podataka iz modela učenika. Svi učenici i nastavni materijal se grupišu u klase sa zajedničkim osobinama (klastere) u cilju odabira preporučenog nastavnog materijala za posebne klase učenika.

Princip klasterovanja podrazumeva minimizovanje sličnosti učenika i nastavnog materijala unutar jedne grupe/kategorije i maksimizovanje razlika između elemenata koji pripadaju različitim grupama/kategorijama. Takvi klasteri se formiraju u sistemu Protus 2.0 sa ciljem odabira odgovarajućeg nastavnog materijala koji će biti prikazan učeniku.

Protus 2.0 sadrži različite verzije nastavnog materijala prilagođenog različitim kategorijama učenika. U toku personalizacije sadržaja, sistem bira odgovarajući nastavni materijal koji će biti prikazan trenutno aktivnom učeniku na osnovu podataka iz njegovog modela. Sistem takođe sakriva napredni materijal od novih učenika ili prikazuje dodatni materijal naprednijim učenicima. Funkcionisanje sistema se zasniva na činjenici da će učenici sa različitim stilovima učenja lakše savladati sadržaj ako im se on prezentuje na odgovarajući način, na primer uz pomoć blok dijagrama ili grafičkog prikaza sintaksnih pravila umesto uz pomoć tekstualnih opisa.

- **Prilagođavanje korisničkog interfejsa** (eng. *user interface adaptation*) predstavlja prilagođavanje prikaza pojedinih elemenata korisničkog interfejsa koje se prikazuju na veb stranicama u cilju preporuke nastavnog materijala.

U okviru sistema Protus 2.0 prilagođavanje korisničkog interfejsa se realizuje prilagođavanjem linkova (eng. *link adaptation*). Sistem modifikuje izgled i/ili uklanja i postavlja linkove na veb stranici kursa. Na taj način se učeniku daju napomene koji linkovi vode ka novom, obrađenom ili preporučenom nastavnom materijalu. Moguće je i ukloniti neke linkove sa stranice ako se proceni da oni vode do nastavnog materijala koji je nebitan ili već savladan od strane trenutno aktivnog učenika.

Fukcionisanje sistema se zasniva na pretpostavkama da će lošiji učenici biti zainteresovani za dodatni nastavni materijal i primere, te im se na korisničkom interfejsu mogu naglasiti linkovi koji vode ka preporučenom nastavnom materijalu.



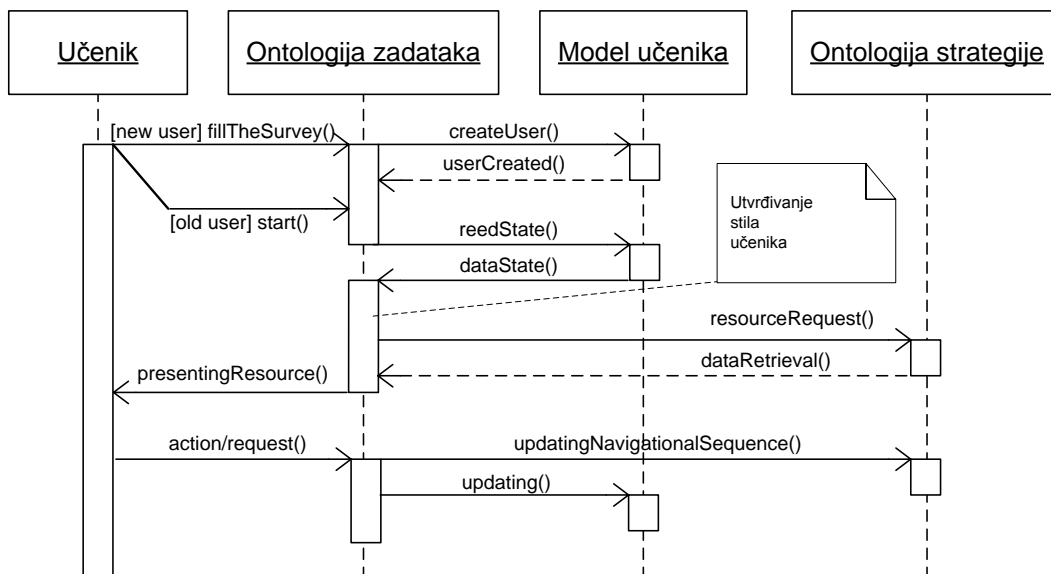
Moguće je koristiti nekoliko pristupa za personalizaciju nastavnog materijala u tutorskom sistemu. Sistem za učenje programskog jezika realizovan u sistemu Protus 2.0 nudi tri tipa personalizacije svakom pojedinačnom učeniku: personalizacija na osnovu stilova učenja (Klašnja-Milićević et al. 2011b), personalizacija uz upotrebu sistema za generisanje preporuka (Klašnja-Milićević et al. 2011a) i personalizacija na osnovu izbora optimalne sekvence nastavnog materijala (Vesin et al., 2012b).

#### 7.4.1 Adaptacija u sistemu Protus 2.0 na osnovu stilova učenja

Različiti učenici imaju različite navike, potrebe i pristupe učenju koji se nazivaju individualni stilovi učenja. Stoga, veoma je važno prilagoditi okruženja i sisteme za učenje tim različitim stilovima učenja u cilju što efikasnijeg procesa učenja. Stilovi učenja se definišu kao jedinstvene navike kojima se učenici koncentrišu, procesiraju, prihvataju i usvajaju informacije i nova znanja (Pritchard, 2013).

Termin *stilovi učenja* predstavlja koncept prepoznavanja razlika između učenika na osnovu načina prikaza ili tipa nastavnog materijala kojim se postiže najbolji efekat razumevanja gradiva (Pashler et al., 2009). Iako postoji više modela određivanja stila učenja, u sistemu Protus 2.0 je implementiran jedan od najčešće korišćenih modela Felder & Silverman (1988) koji predstavlja različite kognitivne stilove učenja učenika (eng. *cognitive styles of learning*) (Klašnja-Milićević et al. 2011b). Na osnovu tih stilova, korisnički interfejs sistema se automatski prilagođava ali se istovremeno učeniku omogućava izbor karakteristika radnog okruženja.

U nastavku poglavlja predstavljen je proces adaptacije sistema na osnovu identifikovanih stilova učenja u sistemu Protus 2.0.



Slika 59. Adaptacija na osnovu stilova učenja

Na početku korisničke sesije Protus 2.0 šalje zahtev za informacijama o statusu kursa za aktuelnog učenika iz *Ontologije modela učenika* (slika 59). Ti podaci uključuju informacije o trenutnoj lekciji i trenutnim kategorijama individualnog stila učenja za svaki od četiri postojeća domena. Zahtev za odgovarajućim resursima koji će biti prikazani učenicima (na osnovu preuzetih podataka) se šalje *Ontologiji nastavne strategije*. U daljem toku

sesije, sistem prati sve akcije učenika kao i zahteve koje upućuje sistemu i upisuje ih u *Ontologiju modela učenika*.

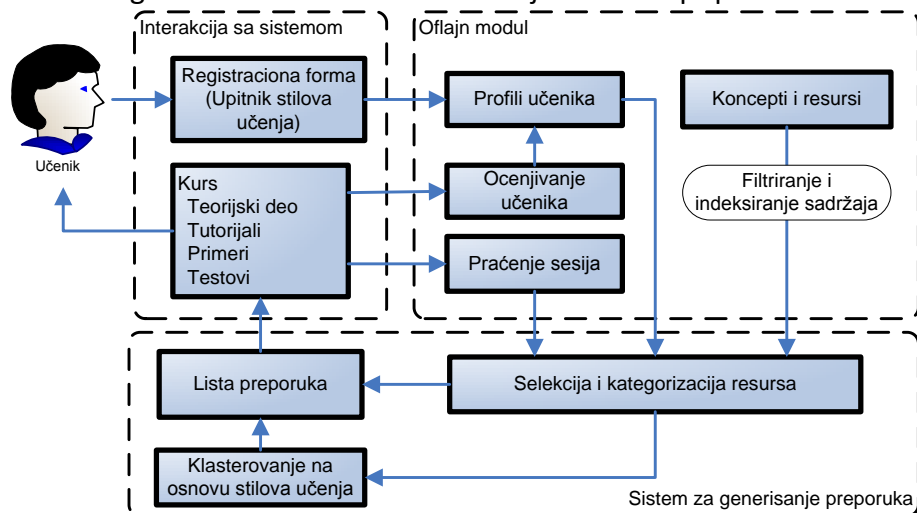
Ako učenik ne prikaže zadovoljavajući nivo znanja tokom izvođenja sesije uz pomoć određenog metoda prezentacije nastavnog materijala za odgovarajući stil učenja, tada sistem modifikuje kategoriju stila učenja kojem učenik pripada. U tom slučaju, sistem menja stil učenja aktivnog učenika u njemu alternativni stil iz istog domena (aktivni u reflektivni, reflektivni u aktivni, osećajni u intuitivni, itd). Utvrđeni prag znanja se može vremenom menjati u zavisnosti od nivoa znanja učenika.

Na primer, ako učenik sa verbalnim stilom učenja, pristupa resursima pojedine lekcije, i tokom te interakcije ne postigne zadovoljavajuće rezultate na odgovarajućem testu za tu lekciju (potreban nivo znanja se čuva u globalnoj promenljivoj *required*), tada se stil učenja tog učenika menja u alternativni stil učenja u okviru istog domena (*domen prijema informacija*), tj. vizuelni stil učenja. Na osnovu toga sledi da će u toku naredne sesije, učeniku biti ponuđeni resursi koji su definisani kao podrška tom novom individualnom stilu učenja.

Detalji procesa adaptacije nastavnog materijala implementirane u sistemu Protus 2.0 kao i pravila adaptacije koja se pritom pokreću su predstavljeni u poglavlju 6.3.1 (pravila adaptacije: PA1 do PA22).

#### 7.4.2 Upotreba sistema za generisanje preporuka

Preporuke nastavnog materijala se u sistemu Protus 2.0 generišu posebno za svakog učenika budući da se i učenici sa istim stilom učenja razlikuju po nivou znanja i sposobnosti rešavanja zadataka (slika 60). Primenjuje se klasterovanje učenika na osnovu individualnih stilova učenja. Izbor nastavnog materijala koji se prikazuje učeniku u nastavku kursa se generiše na osnovu klastera kojem učenik pripada.



Slika 60. Interakcija učenika i sistema Protus 2.0

Sistem za generisanje preporuka se sastoji od tri modula (slika 60):

- Modul interakcije učenika sa sistemom (eng. *learner-system interaction module*) ažurira model učenika. Modul predstavlja korisnički interfejs preko koga se učeniku prezentuje nastavni materijal i prikupljaju informacije o toku kursa trenutno aktivnog učenika. Podaci o aktivnostima učenika (lista posećenih

stranica, rezultati testova, stečene ocene, itd.) se prikupljaju u okviru ovog modula.

- Oflajn modul (eng. *off-line module*) koristi model učenika u cilju utvrđivanja ciljeva učenja i profila učenika. Na osnovu početnog upitnika, određuje se stil učenja pojedinačnog učenika. U nastavku kursa se filtrira nastavni materijal i selektuju adekvatni nastavni resursi koji se prikazuju učenicima u zavisnosti od njihove trenutne pozicije u okviru kursa i identifikovanog stila učenja.
- Sistem za generisanje preporuka (eng. *recommendation module*) generiše listu nastavnog materijala koji se preporučuju učeniku. Sistem generiše niz akcija i preporučuje učeniku nastavni materijal i time prilagođava korisnički interfejs učeniku u okviru trenutne sesije učenja.

Detalji procesa generisanja preporuka implementiranog u sistemu Protus 2.0 kao i pravila adaptacije koja se pritom pokreću su predstavljeni u poglavlju 6.3.3 (pravila adaptacije: PA27 do PA30).

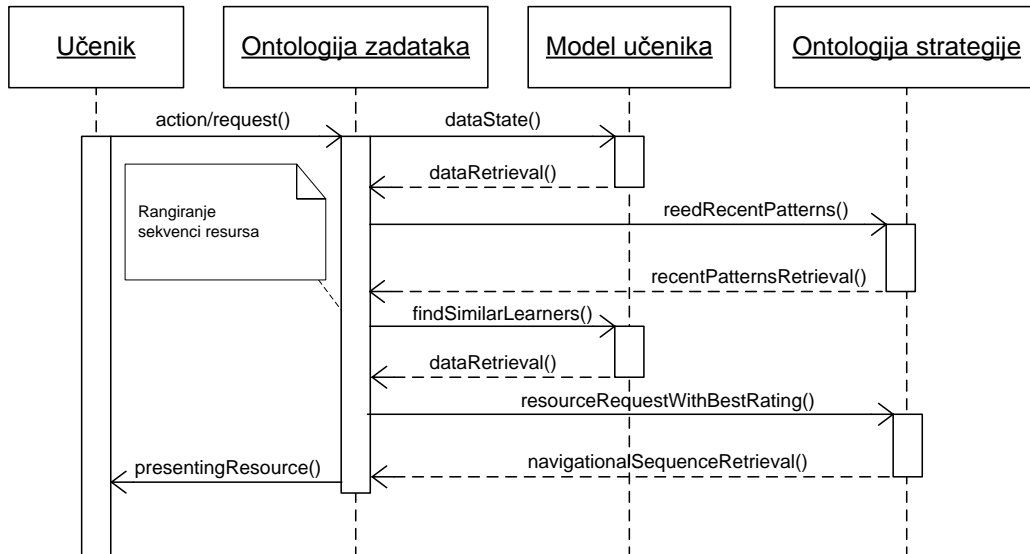
### 7.4.3 Šabloni navigacije

Prilagođavanje rasporeda prikazivanja resursa (eng. *resource sequencing*) predstavlja popularnu tehniku adaptacije u inteligentnim tutorskim sistemima (Janssen et al., 2007). Osnovna ideja je generisanje personalizovanog kursa za svakog učenika uz dinamički izbor optimalnih akcija, prezentacija, primera, zadataka i problema u svakom trenutku pohađanja kursa. Pod optimalnom nastavnom akcijom se podrazumeva operacija koja u trenutnom kontekstu dovodi učenika najbliže nekom definisanom krajnjem cilju učenja. Najčešće taj cilj predstavlja savladavanje date teme i sticanje određenog nivoa znanja u optimalnom vremenskom periodu. Učenici mogu da prave različite putanje kroz nastavni materijal na osnovu njihovih želja i potreba i da generišu različite nizove nastavnih aktivnosti. Sve te varijacije i putanje kroz nastavni materijal se prate, beleže u sistemu i zatim selektuju i sortiraju na osnovu napretka koji su pojedini učenici napravili prateći te putanje.

U cilju praćenja napretka svakog pojedinačnog učenika tokom sesije, sistem beleži rezultate interakcije učenika sa sistemom, stečene ocene i podatke o posećenim konceptima i resursima (navigaciji kroz nastavni materijal). Skup posećenih konceptata i resursa predstavlja šablon navigacije. Tokom korisničkih sesija prati se interakcija svakog učenika sa sistemom i gradi se globalna baza podataka šablona navigacije kroz nastavni materijal.

Nakon svakog zahteva ili akcije učenika, sistem ispituje trenutni raspored poseta resursa i poredi ga sa šablonima navigacije (eng. *Navigational pattern*) ostalih učenika (slika 61). Protus 2.0 pronalazi slične učenike na osnovu njihovih ocena (ostvarenih za tu lekciju) i generiše listu preporuka na osnovu rejtinga tih najčešćih šablona (eng. *frequent sequences*). Šabloni se rangiraju na osnovu rezultata učenika postignutih na testovima (Vesin et al. 2011a). Konkretno, računa se prosek ocena koje učenici ostvare nakon prolaska kroz nastavni materijal u odgovarajućem rasporedu.

Ako je učenik uspešno savladao određeni koncept (tabela 7, poglavlje 7.3.1), koji je podržan određenim resursima (tabela 8, poglavlje 7.3.1) tada sistem dodaje taj resurs u odgovarajući uspešni navigacioni šablon (slika 61).



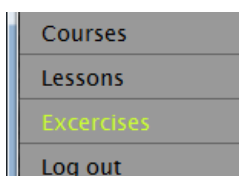
Slika 61. Adaptacija na osnovu navigacije kroz nastavni materijal

Konkretno, to bi značilo da ako je učenik imao interakciju sa sistemom preko određenog šablona navigacije, tada bi sistem trebalo da preporuči i sledeći najbolje rangirani resurs koji odgovara tom šablonu.

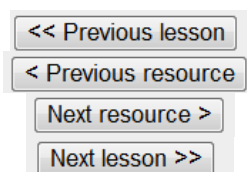
Na primer, neka je učenik posetio resurse za prikaz uvodnih razmatranja (resurs *Intro*), osnovnih informacija (resurs *Basic info*) i objašnjenja (resurs *Explanation*) za neku lekciju. Prilikom generisanja preporuke koji resurs će aktivnom učeniku u nastavku kursa biti prikazan, sistem uzima u obzir učenike koji su u istom rasporedu posetili resurse u okviru te lekcije. Sistem zatim poredi ocene onih učenika koji su u nastavku lekcije posetili resurs sa sintaksnim pravilima (resurs *Syntax rules*) i ocene onih učenika koji su prešli na rešavanje zadataka (poseta resursu *Activity*). U nastavku se učeniku preporučuje resurs koji je u nastavku lekcije doneo bolje rezultate testa učenicima koji su ga posetili.

Preporuka se generiše tako što se atributu *recommendation* izabranog resursa dodeljuje vrednost *true*, te se na osnovu toga izvršava neka od sledećih izmena korisničkog interfejsa (u zavisnosti od tipa resursa):

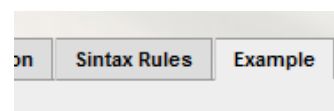
- link ka tom resursu je prikazan ili naglašen (slika 62a),
- elementi korisničkog interfejsa za sekvencijalnu navigaciju će biti prikazani (pružajući mogućnost slobodne navigacije kroz nastavni materijal) ili sakriveni (slika 62b),
- dodavanje odgovarajuće kartice u panelu (eng. *Tabbed pane*) koji prikazuju nastavni materijal koji sadrži dodatne informacije, primere ili objašnjenja gradiva (slika 62c).



a. obeležavanje elemenata



b. Prikaz elemenata za navigaciju



c. Dodavanje elemenata

Slika 62. Adaptacija korisničkog interfejsa

Takođe, ako učenik poseti određeni koncept i tokom te interakcije pristupi rešavanju testa vezanog za taj koncept i dobije pozitivnu ocenu, tada sistem beleži rezultate interakcije učenika sa sistemom u ontologiju modela učenika (listing 1) uz pomoć pravila adaptacije prikazanih u poglavlju 6.3 (pravila adaptacije PA27-PA30) i obeležava taj resurs kao savladan (tabela 8, poglavlje 7.3.1).

**Listing 1. Pseudokod procesa izvođenja kursa u Protusu 2.0**

```
begin ProtusKurs
do
  prikaziKoncept
  odaberiPrviResurs
  do
    if gotovPregled
      testZnanja
      obradiRezultate
    else
      odaberiNaredniResurs
      prikaziResurs
      prilagodiKorisnickiInterfejs
    end if
  azurirajSesiju
  while nijeSavladanKoncept
while nijeSavladanKurs
```

#### 7.4.4 Proces evaluacije znanja učenika

Nakon što učenik prođe kroz nastavne sadržaje u skladu sa svojim stilom učenja, sistem nudi opciju testiranja znanja učenika. Ocene učenika se utvrđuju na osnovu procenta tačnih odgovora, tj ocene su skalirane na sledeći način:

- 5 (odlično) - (80–100%)
- 4 (vrlo dobro) - (70–79%)
- 3 (dobro) - (60–69%)
- 2 (dovoljno) - (50–59%)
- 1 (nedovoljno) - (0–49%)

Dva učenika se smatraju sličnim ako ih je sistem isto ocenio za isti niz savladanog nastavnog materijala. Proces generisanja preporuka se u nastavku procesa učenja odvija na osnovu ostvarenih ocena i tehnike *collaborative filtering*-a (opisanog u poglavlju 3.3). Ova tehnika omogućava da se učenici na osnovu ocena koje su ostvarili tokom učenja grupišu po sličnosti u klastere.

#### 7.5 Upotreba i funkcionisanje sistema

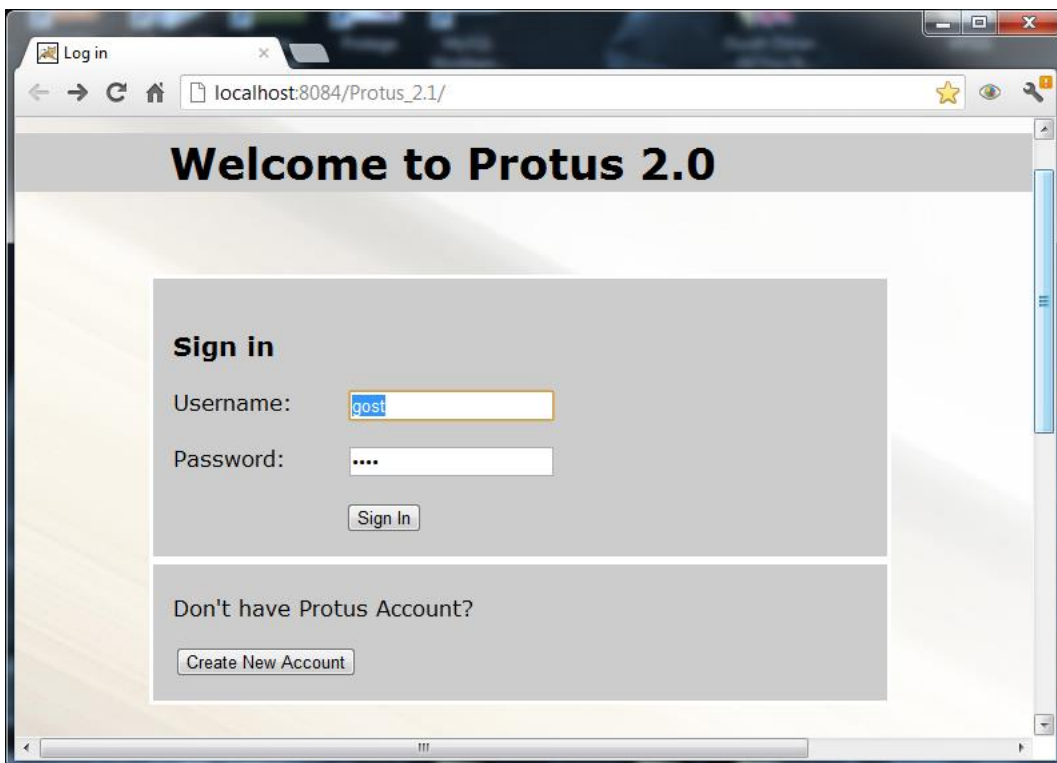
Protus 2.0 je namenjen razvoju kurseva iz različitih domena. U ovom momentu razvijen je nastavni materijal za učenje programskog jezika Java i sistem u potpunosti podržava ovaj kurs.

On nudi učenicima veb interfejs za pregled nastavnog materijala i testiranje znanja. Kreirane su različite verzije nastavnih resursa za 18 lekcija iz programiranja u programskom jeziku Java namenjenih učenicima sa različitim stilovima učenja.

Početa stranica sistema nudi mogućnost registracije novog učenika ili prijavljivanje već registrovanog učenika (slika 63). Novi učenik se prijavljuje preko odgovarajuće forme čime se kreira početni profil učenika (slika 64). Podaci o učenicima se čuvaju u ontologiji modela učenika i uključuju:

- osnovne lične podatke koje unese učenik kao što su: ime, prezime, korisničko ime, lozinka i godište (statičke informacije) i
- informacije o toku kursa, stilovima učenja, stečenim ocenama, itd (dinamičke informacije).

Standardizovani oblik modela učenika prikazanog u ovoj disertaciji omogućava razmenu podataka o učenicima sa drugim sistemima.



Slika 63. Korisnička forma za prijavljivanje učenika

Nakon registracije, sistem nudi učeniku upitnik (slika 65) koji sadrži 44 pitanja za utvrđivanje inicijalnog stila učenja preko ILS (eng. *Index of learning styles*) upitnika koje su definisali Felder & Soloman (1996).

Prilog 3 disertacije sadrži kompletnu listu pitanja ILS upitnika.

Početni stil učenja se izračunava na osnovu odgovora učenika i beleži se u ontologiju modela učenika koja se koristi za dalju adaptaciju sistema konkretnom pojedinačnom učeniku (Klašnja-Milićević et al. 2011b). Stil učenika odražava način prezentacije nastavnog materijala koji njemu najviše odgovara.

Od velikog je značaja da učenici ozbiljno shvate upitnik i odgovore na pitanja koja se u njemu nalaze, jer od njihovih odgovora direktno zavisi inicijalizacija modela učenika. Protus 2.0 onemogućava početak kursa dok učenik ne odgovori na sva pitanja.

Registration

Please, provide us with the basic information:

First name:  \*

Last name:  \*

Year of birth:  \*

Username:  \*

Password:  \*

Re-enter password:  \*

\* required field

Slika 64. Registraciona forma sistema Protus 2.0

**Please, fill the questionnaire first**

For each of the 44 questions below select either "a" or "b" to indicate your answer. Please choose only one answer for each question. If both "a" and "b" seem to apply to you, choose the one that applies more frequently. When you are finished selecting answers to each question please select the submit button at the end of the form.

- I understand something better after I
  - try it out
  - think it through
- I would rather be considered
  - realistic
  - innovative
- When I think about what I did yesterday, I am most likely to get
  - a picture
  - words
- I tend to
  - understand details of a subject but may be fuzzy about its overall structure
  - understand the overall structure but may be fuzzy about details
- When I am learning something new, it helps me to
  - talk about it
  - think about it
- If I were a teacher, I would rather teach a course
  - that deals with facts and real life situations
  - that deals with ideas and theories

Slika 65. ILS upitnik

### 7.5.1 Bodovanje upitnika i određivanje početnog stila učenja

Svako pitanje iz upitnika pripada jednoj od četiri dimenzije stilova učenja (obrada informacija, percepcija informacija, prijem informacija i razumevanje informacija). Svako od četiri dimenzije je dodeljeno po 11 pitanja. Za svako pitanje su ponuđena dva odgovora od kojih svaki usmerava krajnji rezultat ka jednoj od dve kategorije u okviru odgovarajuće dimenzije stilova učenja.

Nakon što učenik selektuje sve odgovore, sistem popunjava odgovarajuću tabelu čija forma je prikazana u tabeli 12. Za svaki odgovor upisuje se po jedan poen u odgovarajuća polja tabele. Zatim se sabiraju brojevi u odgovarajućim kolonama. Krajnji indeks stila učenja u okviru jedne dimenzije se određuje na osnovu broja odgovora pod *a* i pod *b*.

Tako je vrednost indeksa -6 ako su svi odgovori na jedanaest pitanja tipa *a*, indeks je -5 ako je deset odgovora tipa *a* i samo jedan odgovor tipa *b*, indeks je -4 ako je devet odgovora tipa *a* i dva odgovora tipa *b*, itd.

#### Tumačenje indeksa stila učenja

Ako je izračunati indeks stila učenja u rasponu od -2 do 2 učenik neznatno naginje jednoj ili drugoj kategoriji u okviru dimenzije. Ukoliko indeks iznosi -4, -3, 3 ili 4, tada učenik umereno naginje jednoj od kategorija i on će lakše učiti u okruženju koje favorizuje tu kategoriju. U slučaju da je indeks -6, -5, 5 ili 6 tada je učenik jako naklonjen jednoj kategoriji skale i može imati velike teškoće ako bi učio u okruženju koje podržava suprotnu kategoriju.

Tabela 12. Primer popunjene tabele za određivanje stilova učenja

Aktivni/refleksivni			Osećajni/Intuitivni			Vizuelni/Verbalni			Globalni/sekvencijalni		
Pitanje	a	b	Pitanje	a	b	Pitanje	a	b	Pitanje	a	b
1	1		2	1		3		1	4	1	
5		1	6	1		7	1		8		1
9		1	10		1	11		1	12	1	
13		1	14		1	15	1		16	1	
17	1		18		1	19	1		20	1	
21	1		22		1	23		1	24		1
25		1	26	1		27	1		28		1
29	1		30		1	31		1	32	1	
33	1		34	1		35		1	36	1	
37	1		38		1	39	1		40	1	
41	1		42		1	43	1		44	1	
Ukupno (zbir jedinica u svakoj koloni)											
Aktivni/refleksivni			Osećajni/Intuitivni			Vizuelni/Verbalni			Globalni/sekvencijalni		
	a	b		a	b		a	b		a	b
Zbir	7	4	Zbir	4	7	Zbir	6	5	Zbir	8	3
Indeks odgovarajućeg stila											
-2			2			-1			-3		

Inicijalizacija modela učenika ima važnu ulogu prilikom definisanja početnih opcija personalizacije u sistemu.

U listinzima 2-4 su prikazani segmenti koda koji na osnovu odgovora učenika izračunavaju numeričku vrednost stilova učenja za učenika i inicijalizuje odgovarajući model učenika.



Sistem Protus 2.0 uz pomoć segmenta koda na listingu 2. čita odgovore koje je učenik uneo za pitanja iz upitnika i unosi ih u odgovarajući niz.

### Listing 2. Učitavanje odgovora učenika

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    int odgovori[] = new int[44];
    Styles stil = new Styles();

    for (int i = 0; i < odgovori.length; i++) {
        try {
            if (request.getParameter("RadioGroup"+(i+1)).equals("radio1")) {
                odgovori[i]=0;
            } if (request.getParameter("RadioGroup"+(i+1)).equals("radio2")) {
                odgovori[i]=1;
            }
        } catch (java.lang.NullPointerException e) {
            stil.setPoruka("Mora se dati odgovor na sva pitanja");
            request.setAttribute("stil", stil);
            request.getRequestDispatcher("Questionnaire.jsp").forward(request, response);
        }
    }
}
```

Zatim se sabiraju odgovori koji pripadaju jednoj kategoriji stila učenja u okviru svake dimenzije (Listing 3).

### Listing 3. Sabiranje odgovora po grupama

```
int a = 0, b = 0, c = 0, d = 0;
for (int i = 0; i < odgovori.length; i++) {
    if (i%4==0) {
        if(odgovori[i]==1){
            a++;
        }
    }else if (i%4==1){
        if(odgovori[i]==1){
            b++;
        }
    }else if (i%4==2){
        if(odgovori[i]==1){
            c++;
        }
    }else if (i%4==3){
        if(odgovori[i]==1){
            d++;
        }
    }
}
}
```

Na osnovu tih podataka izračunavaju se konačni indekse za svaku od četiri dimenzije stilova učenja (Listing 4).

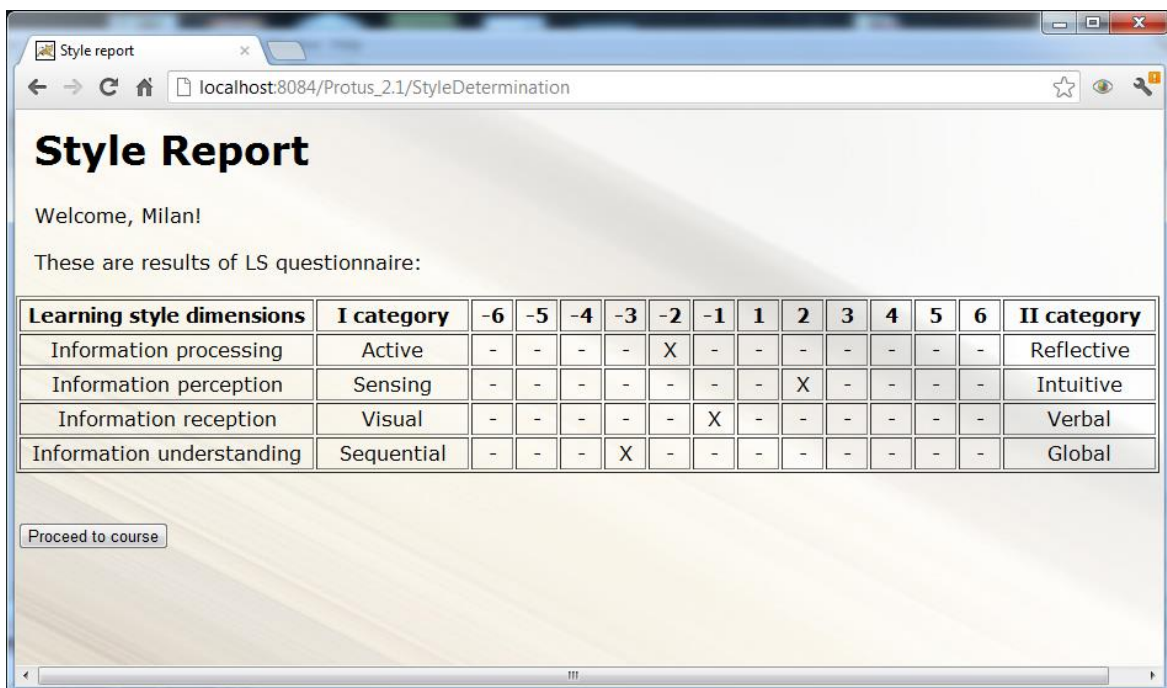
**Listing 4. Izračunavanje konačnih indeksa stilova učenja**

```

if (a>5) {
    stil.setProcessing(((2*a-11)+1)/2);
} else {
    stil.setProcessing(((2*a-11)-1)/2);
}
if (b>5) {
    stil.setPercepcion(((2*b-11)+1)/2);
} else {
    stil.setPercepcion(((2*b-11)-1)/2);
}
if (c>5) {
    stil.setReception(((2*c-11)+1)/2);
} else {
    stil.setReception(((2*c-11)-1)/2);
}
if (d>5) {
    stil.setUnderstanding(((2*d-11)+1)/2);
} else {
    stil.setUnderstanding(((2*d-11)-1)/2);
}

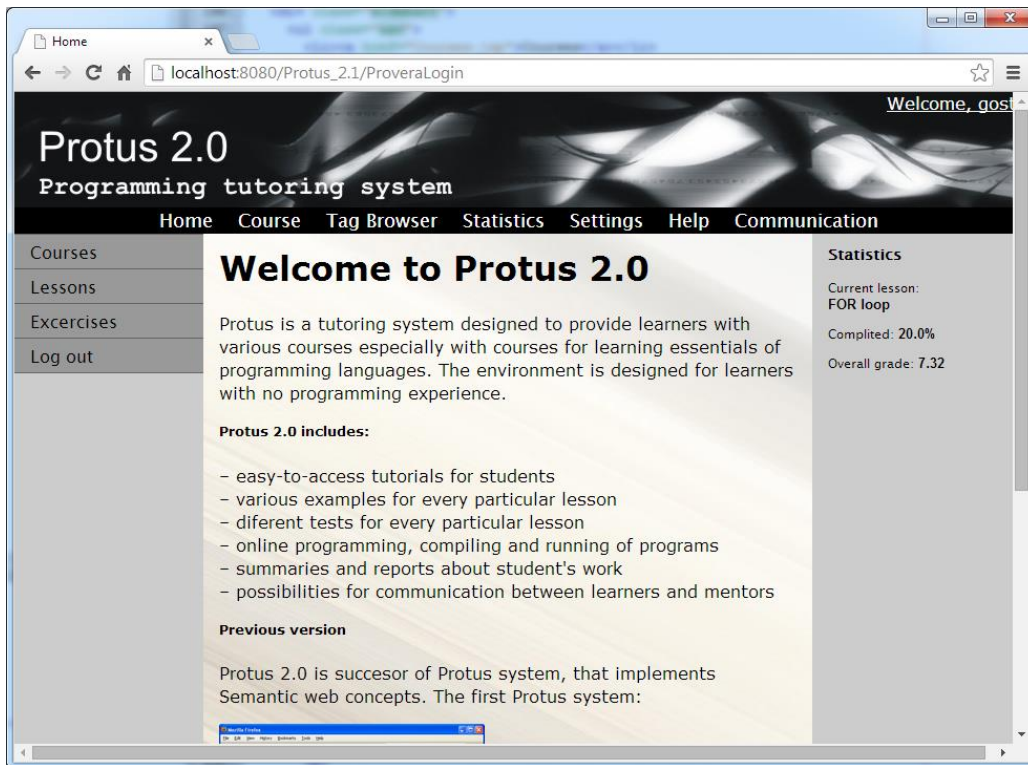
```

Na osnovu dobijenih podataka Protus 2.0 ima kompletnu informaciju o početnom stilu učenika, ažurira *Ontologiju modela učenika* i prikazuje rezultate upitnika na narednoj stranici (slika 66).



**Slika 66. Izveštaj o rezultatima upitnika**

Nakon uspešne registracije novog učenika, inicijalizuje se početna sesija za njega. Svi podaci o trenutnoj lekciji, procentu pređenog nastavnog materijala, oceni koju je učenik ostvario u toku sesije, prosečnoj oceni koju je učenik ostvario tokom celog kursa, itd. se postavljaju na inicijalne vrednosti i unose u odgovarajuću *Ontologiju modela učenika*. Na taj način učenik započinje konkretno učenje nastavnog materijala za izabrani kurs (slika 67)



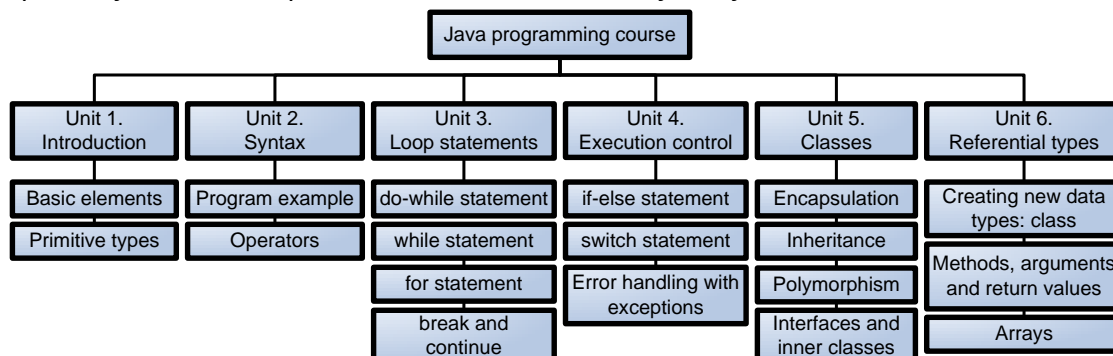
Slika 67. Početna stranica sistema Protus 2.0

### 7.5.2 Realizacija kursa za učenje osnova Java programskog jezika

Nastavni materijal za programski jezik Java u sistemu Protus 2.0 je organizovan u 6 jedinica sa ukupno 18 lekcija (slika 68). Lekcije su raspoređene u sledeće jedinice:

- uvod (eng. *Introduction*),
- sintakstna pravila (eng. *Syntax*),
- petlje (eng. *Loop statements*),
- kontrola toka programa (eng. *Execution control*),
- klase (eng. *Classes*),
- referencijalni tipovi (eng. *Referential types*).

Nakon prijave učenika u sistem (slika 63), korisnička sesija se inicijalizuje na osnovu specifičnih podataka o učeniku koji se čuvaju u ontologiji modela učenika i niz resursa mu se prikazuje na osnovu podataka o trenutno aktivnoj lekciji i resursu.

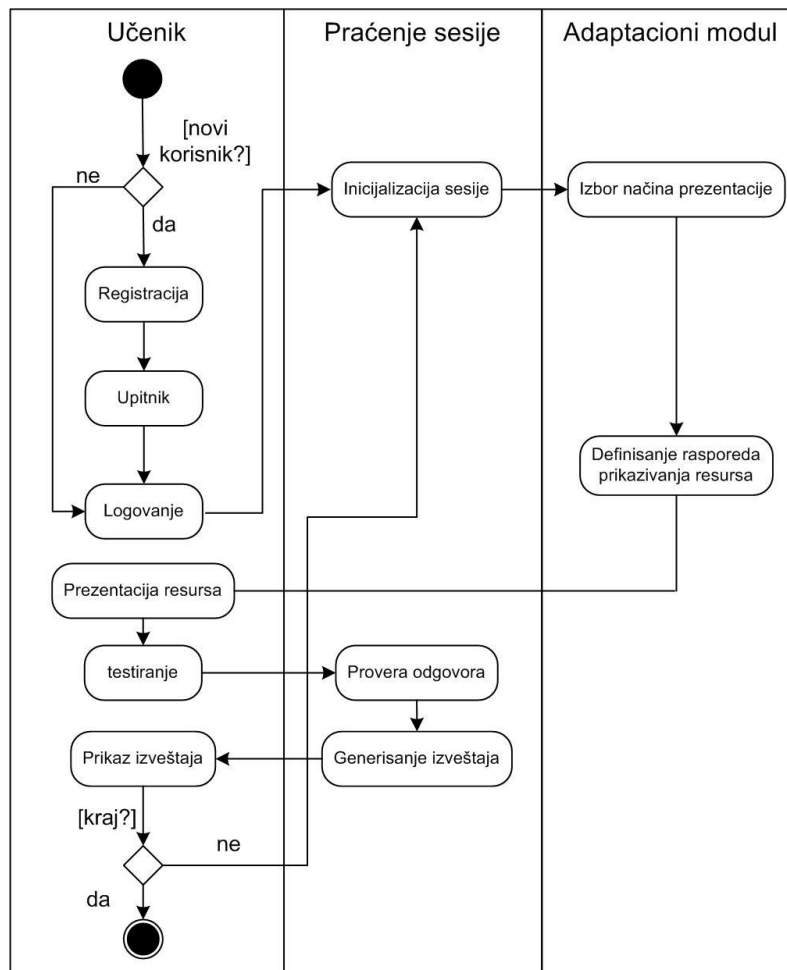


Slika 68. Hijerarhija lekcija u kursu iz Java programskog jezika

U toku učenja i korišćenja sistema učeniku se lekcije mogu prikazivati na dva načina:

- Prikaz lekcija u tačno određenom rasporedu. Ovakav tip prikaza se nudi učenicima sa sekvencijalnim stilom učenja.
- Učenik ima mogućnost da sam bira redosled prikaza lekcija. Učenicima sa globalnim stilom učenja se omogućuje samostalan izbor redosleda prikaza lekcija i resursa.

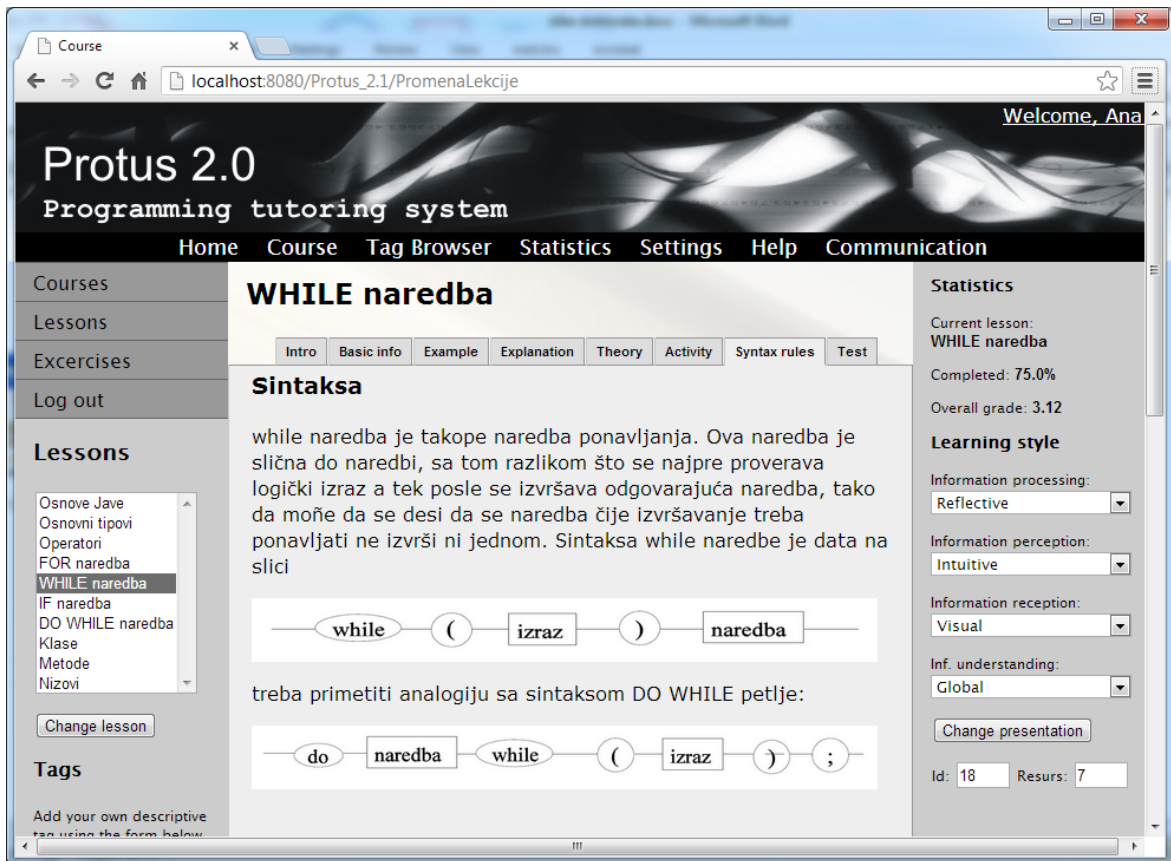
Nakon izbora konkretne lekcije koja se prikazuje učeniku, sistem bira način prezentacije nastavnog materijala na osnovu aktuelnog stila učenja i vrši izmene na elementima korisničkog interfejsa (slika 69).



Slika 69. Proces učenja u sistemu Protus 2.0

Korisnički interfejs za učenika je predstavljen na slici 70. Svaka lekcija je predstavljena nizom kartica od kojih svaka sadrži odgovarajući deo lekcije, tj. prikazuje odgovarajući nastavni resurs.

U zavisnosti od generisanih preporuka menja se raspored kartica, prikazuju se ili sakrivaju pojedine opcije, bira se tip prikaza resursa ili se naglašavaju pojedini linkovi.

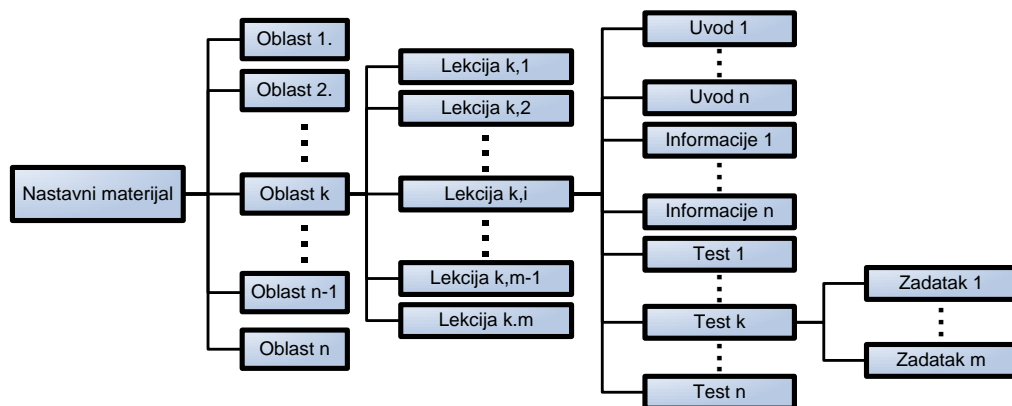


Slika 70. Korisnički interfejs za učenika

## 7.6 Nastavni materijal u sistemu Protus 2.0

Protus 2.0 predstavlja opšti tutorski sistem koji se može koristiti i primenjivati za različite kurseve. Uprkos činjenici da je sistem zamišljen kao opšti, prva u potpunosti implementirana i testirana verzija se koristi za učenje osnova programskog jezika Java (Klašnja-Milićević, 2011a). Java je odabrana zato što je popularan programski jezik na Univerzitetu u Novom Sadu i što je jasan primer objektno-orijentisanog programskog jezika i kao takav je pogodan za učenje koncepata objektno-orijentisanog stila.

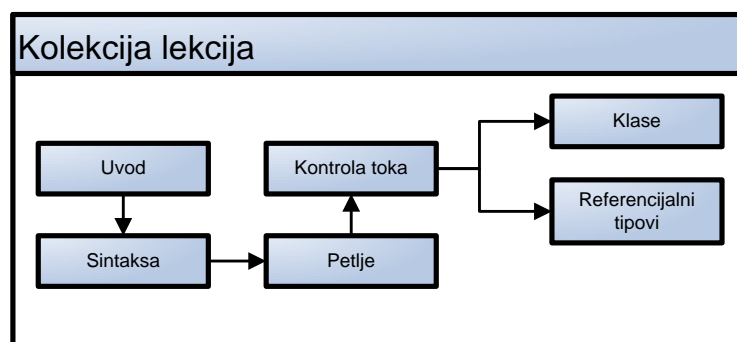
Protus 2.0 je interaktivni sistem sa osnovnim ciljem da učenicima omogući upotrebu nastavnog materijala u okviru početnog programerskog kursa.



Slika 71. Hijerarhija nastavnog materijala

Nastavni materijal je podeljen na oblasti (eng. *Units*), od kojih svaka može da sadrži više lekcija (konceptata) što je i prikazano na slici 71. Svaka lekcija sadrži resurse (stranice) različitog tipa. Tako imamo resurse koji sadrže: uvodne napomene (eng. *intro*), osnovne informacije (eng. *basic info*), primere (eng. *example*), objašnjenja (eng. *explanation*), teoriju (eng. *theory*), zadatke za rešavanje (eng. *activity*), sintakсна pravila (eng. *syntax rules*), itd.

Svakoј lekciji se može dodeliti proizvoljan broj resursa različitog tipa. Resursi koji prikazuju objašnjenja i sintakсна pravila imaju po dve varijante prikaza (verbalni i tekstualni) koji se biraju na osnovu odgovarajuće kategorije stila učenja iz dimenzije *Prijema informacija* trenutnog učenika.



Slika 72. Grupisanost lekcija po jedinicama

Nastavni materijal u sistemu Protus 2.0 je realizovan u formi *html* dokumenata. Sistem na osnovu podataka iz modela učenika, učitava dokument koji je određen za prikaz u datom trenutku za aktivnog učenika i prikazuje ga u *web browser*-u.

Lekcije (koncepti) implementiranog kursa programskog jezika Java su grupisane u 6 nastavnih jedinica (eng. *units*) kako je prikazano na slici 72.

Na primeru lekcije *If naredbe* ilustrovani su resursi koji prikazuju nastavni materijal. Prikazani su resursi za uvodni deo (slika 73), osnovne informacije (slika 74), primeri (slika 75), objašnjenja (slika 76), sintakсна pravila za učenika sa verbalnim stilom učenja (slika 77) i sintakсна pravila za učenika sa vizuelnim stilom učenja (slika 78).

Slika 73. Resurs za prikaz uvoda lekcije *If naredba*

## IF naredba

Intro Basic info Example Explanation Theory Activity Syntax rules Test

### If naredba - osnovne informacije

if naredba utvrđuje istinitost izraza u zagradi i ako je tačna izvršavaju se zadate akcije

```
if ( x < 10 ) x = 10;
```

Ako je vrednost promenljive x manja od deset tada se promenljivoj dodeljuje vrednost 10.

Moguće je zapisati i:

```
if ( x < 10 )  
x = 10;
```

Ili:

```
if ( x < 10 ) { x = 10; }
```

Slika 74. Resurs za prikaz osnovnih informacija lekcije *If naredba*

## IF naredba

Intro Basic info Example Explanation Theory Activity Syntax rules Test

### If naredba - Primer 1

Primer dela programa koji ispisuje veći od dva broja

```
int x = 5;  
int y = 7;  
if ( x > y ) {  
    System.out.println(x + " je veci od " + y);  
} else {  
    System.out.println(x + " nije veci od " + y);  
}
```

### If naredba - Primer 2

Primer dela programa koji ispisuje bonus koji se isplaćuje na platu ako ona iznosi preko 75000 dolara

```
if (prodaja > 75000) {  
    bonus = 1000;  
    System.out.println("Bonus iznosi $1000");  
}
```

Slika 75. Resurs za prikaz primera lekcije *If naredba*

## IF naredba

Intro Basic info Example Explanation Theory Activity Syntax rules Test

### If naredba - objašnjenje

If naredba je korisna za izbor između više mogućnosti:

```
if ( n == 1 ) {  
    // izvršavanje prvog bloka naredbi  
} else if ( j == 2 ) {  
    // izvršavanje drugog bloka naredbi  
} else {  
    //ako ni jedan od predhodnih uslova nije //zadovoljen  
}
```

<< Previous lesson < Previous resource Next resource > Next lesson >>

Slika 76. Resurs za prikaz objašnjenja lekcije *If naredba*

## IF naredba

Intro Basic info Example Explanation Theory Activity **Syntax rules** Test

### If naredba - sintaksa

If...else izraz sintaksa:

```

if(uslov){
    izrazi;
}else{
    izrazi;
}

```

if ... else izraz ispituje tačnost izraza. Izvršava se jedan niz naredbi ako je vrednost true ili drugi niz naredbi ako je vrednost false.

```

if (x != oldx) {
    System.out.print("x je promenjena");
else {
    System.out.print("x nije promenjena");
}

```

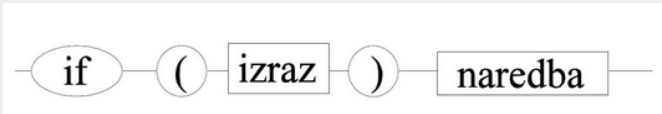
Slika 77. Resurs za prikaz sintakasnih pravila za učenika sa verbalnim stilom učenja

## IF naredba

Intro Basic info Example Explanation Theory Activity **Syntax rules** Test

### IF naredba - sintaksa

Sintaksa if naredbe je data na slici. if naredba se sastoji od ključne reči if, izraza u zagradama koji mora biti tipa boolean i proizvoljne Java naredbe.



Naredba if se izvršava tako što se najpre izračuna vrednost izraza. Ako je vrednost izraza true, izvršava se naredba navedena iza izraza. U protivnom, ako je vrednost izraza false, navedena naredba se ne izvršava. Takođe treba primetiti da ukoliko postoji više naredbi koje treba da se izvrše ako je uslov tačan onda se one moraju staviti u poseban blok.

### if else naredba

Slika 78. Resurs za prikaz sintakasnih pravila za učenika sa vizuelnim stilom učenja

Kada učenik savlada određenu količinu nastavnog materijala vezanog za jednu lekciju (koncept), sistem nudi test za proveru znanja učenika koji se sastoji iz više zadataka. Svako lekciji se može dodati proizvoljan broj zadataka. Protus 2.0 iz baze podataka bira odgovarajući broj zadataka vezanih za datu lekciju, formira test i prikazuje ga učeniku (Vesin et al. 2012b).

Primer testa za lekciju *for petlja* je dat na slici 79. Nakon odgovora učenika na zadatke testa, Protus 2.0 prikazuje rezultate testiranja i tačne odgovore. Ocene koje učenici dobijaju se izračunavaju na osnovu procenta tačnih odgovora i beleže se u *ontologiju modela učenika*.



## FOR naredba

Intro Basic info Example Explanation Theory Activity Syntax rules **Test**

### FOR naredba - Test

1. Sintaksa kojeg od sledećih zaglavlja for petlje je ispravna:

- for (int k > 0; k = 10; i++){}
- for (int k = 0, k < 10, i = i + 1){}
- for (int k = 0; k < 10; i++){}
- for (k = 0; k < 10; i = ++){}

2. Sintaksa kojeg od sledećih zaglavlja for petlje je ispravna:

- for (int = 0; i < 10; i++){}
- for ( i < 10; int=0; ++i){ }
- for (int i = 0; i= i + 1; i ++ ){}
- for (int i = 0; i < 9; i ++ ){}

3. Koja je vrednost promenjive j nakon izvršavanja naredne for petlje

```
int j = 1;
for (int i = 0; i < 5; i++){
    j *= i;
}
```

- 16
- 120
- 24
- 0

4. Koja je vrednost promenjive j nakon izvršavanja naredne for petlje

```
int j = 0;
for (int i = 1; i < 4; ++i){
    j = i + j;
}
```

- 6
- 10
- 5
- 9

<< Previous tasks

Next tasks >>

Clear test

Quit test

Slika 79. Prikaz testa lekcije *For petlja*

U svakom trenutku korisničke sesije učenik može da izabere odgovarajuću opciju i dobije kratak pregled stanja kursa kojeg pohađa i prikaz svojih stilova učenja (slika 80). Sličan izveštaj se prikazuje i nakon što učenik odjavi trenutnu sesiju.

## Statistics

### Overall data:

Current Lesson: **Metode**  
Course completed: **45.0%**  
Overall grade: **4.41**  
Overall time: **02:31:02**

### Current learning styles:

Information processing: **Weak Active (koeficient: -1)**  
Information perception: **Weak Sensing (koeficient: -1)**  
Information reception: **Weak Visual (koeficient: -1)**  
Inf. understanding: **Weak Sequential (Koeficient: 1)**

Slika 80. Izveštaj o stanju kursa i trenutnim stilovima učenja

Učeniku se još nude mogućnosti pregleda ponuđenih kurseva (trenutno je ponuđen samo jedan) (slika 81), kratko uputstvo korišćenja sistema (slika 82), slanja poruka mentoru (slika 83), pregled lekcija aktivnog kursa (slika 84), pregleda unetih tagova za konkretnu lekciju (slika 85) i podešavanja osnovnih opcija sistema (slika 86).

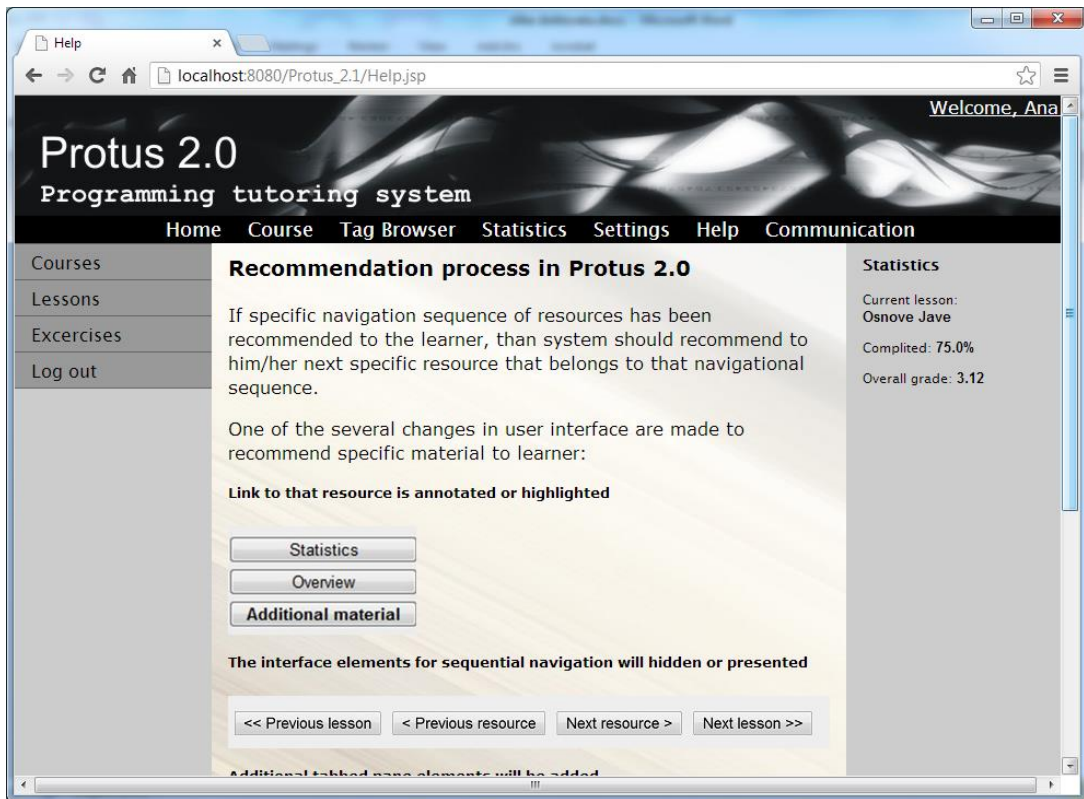
The screenshot shows a web browser window displaying the Protus 2.0 Programming tutoring system. The page title is "Protus 2.0 Programming tutoring system". The navigation menu includes Home, Course, Tag Browser, Statistics, Settings, Help, and Communication. The main content area is titled "Courses We Are Offering" and contains the following information:

- Below please find the courses that we are offering in Protus 2.0. For a full listing of all the courses in the curriculum use the drop-down menu:
- List of courses: Java programming course
- Units: 6
- Lessons: 17
- Learning styles supported: All
- Number of students currently attending: 0
- Course structure: A hierarchical diagram showing the "Java programming course" branching into six units: Unit 1: Introduction, Unit 2: Syntax, Unit 3: Loop statements, Unit 4: Execution control, Unit 5: Classes, and Unit 6: Referential types.

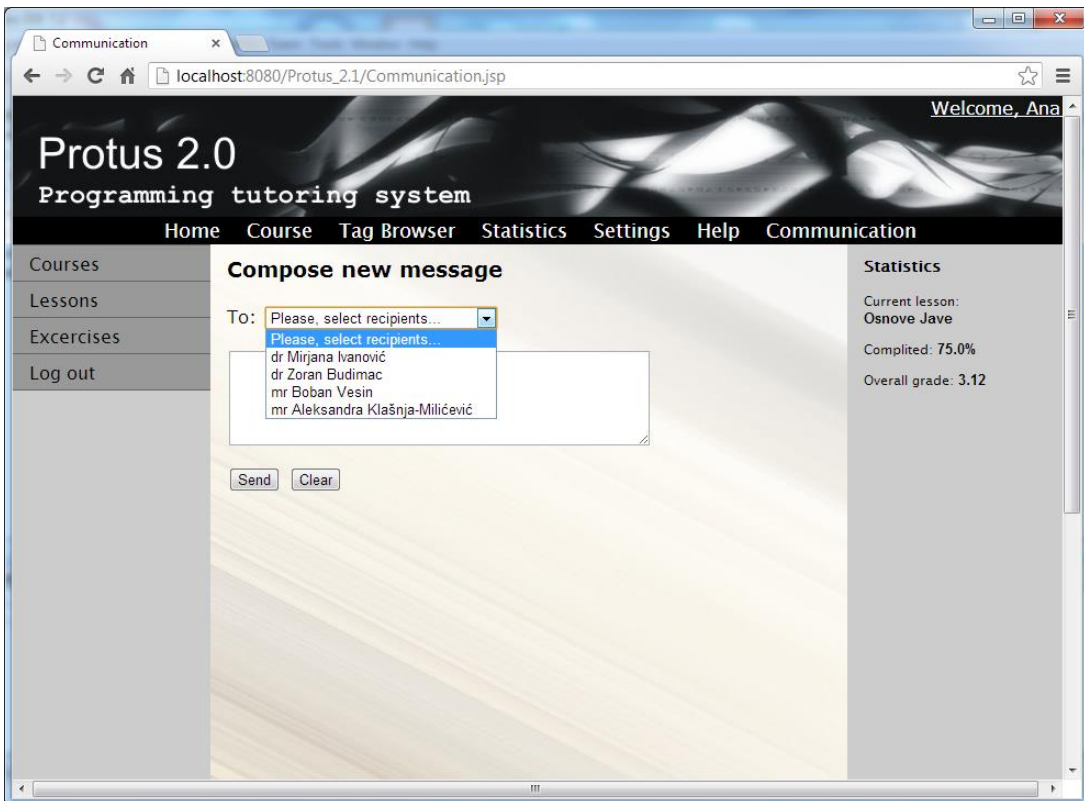
On the right side, there is a "Statistics" sidebar with the following data:

- Current lesson: FOR naredba
- Completed: 45.0%
- Overall grade: 4.41

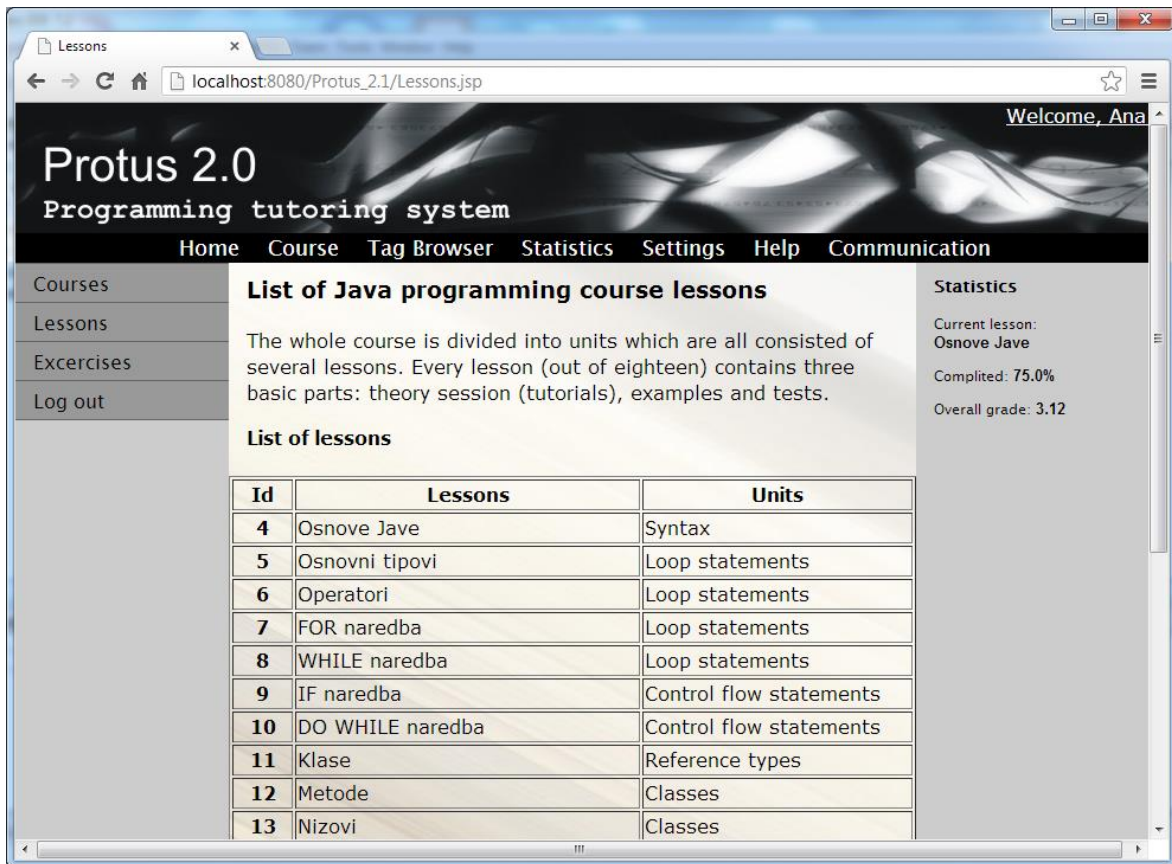
Slika 81. Pregled ponuđenih kurseva



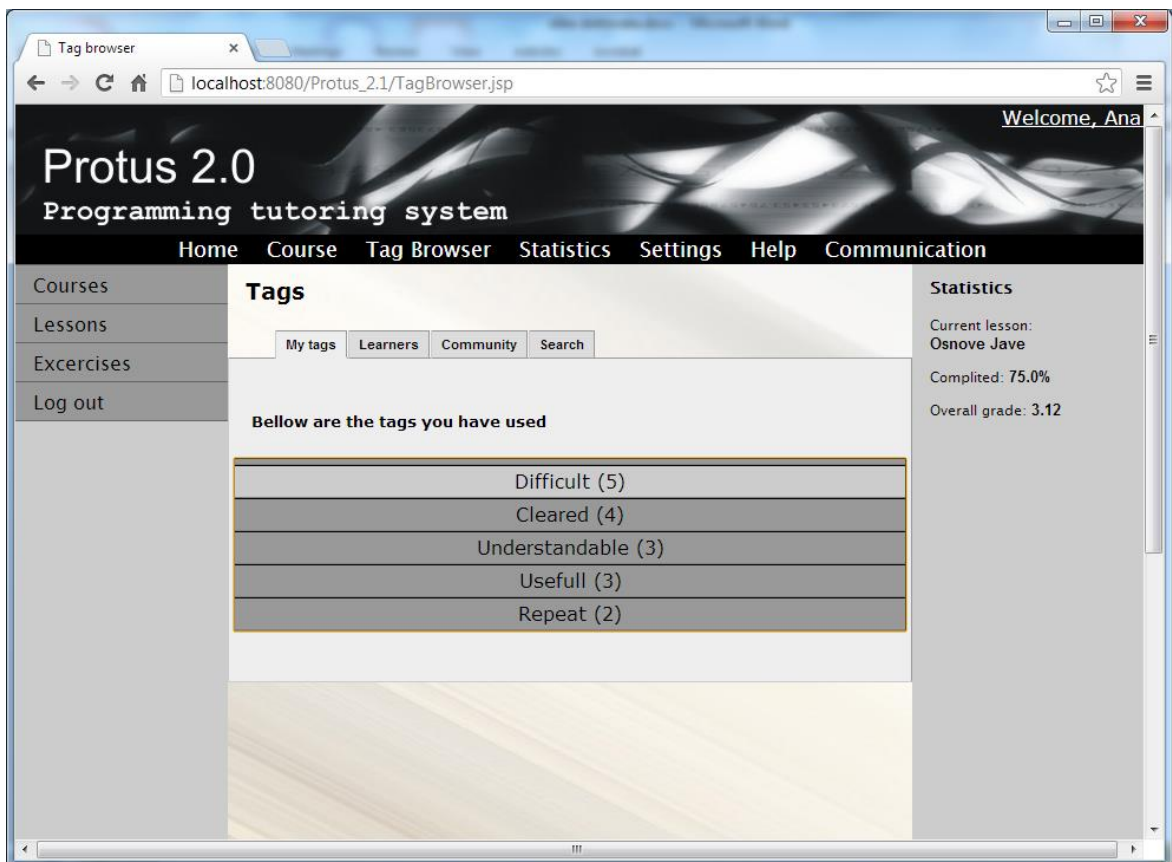
Slika 82. Uputstvo korišćenja sistema



Slika 83. Opcije komunikacije sa mentorom

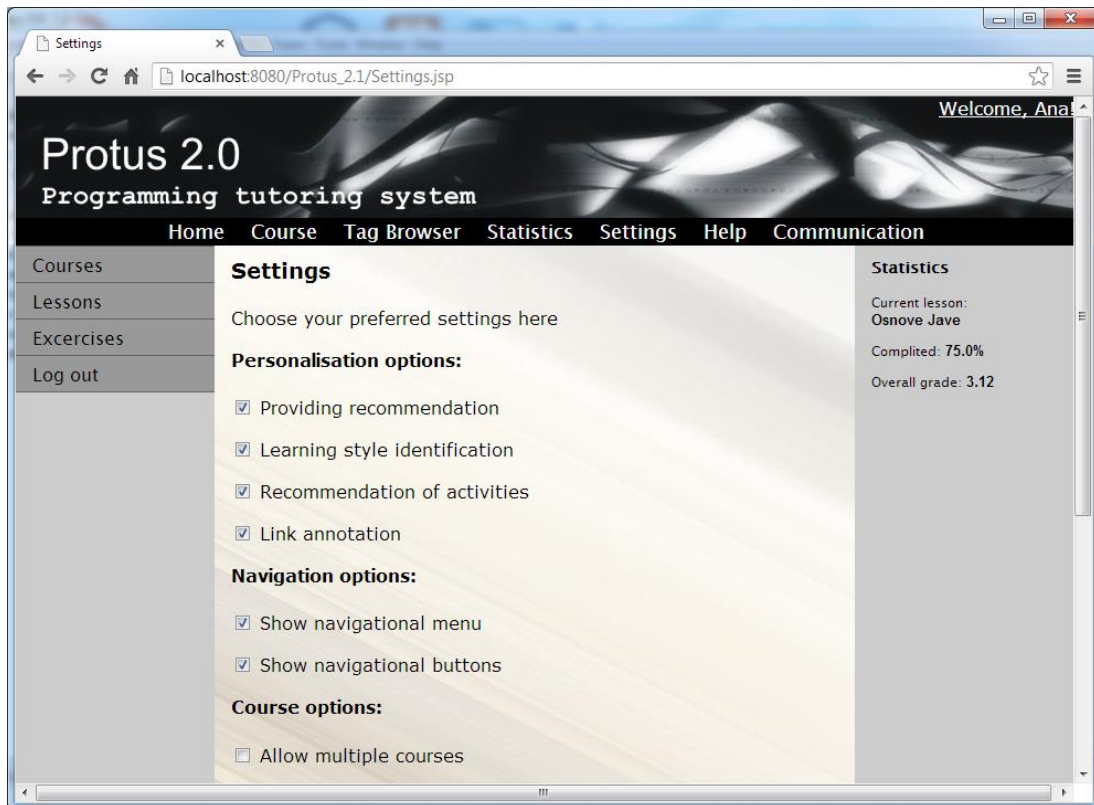


Slika 84. Pregled lekcija aktivnog kursa



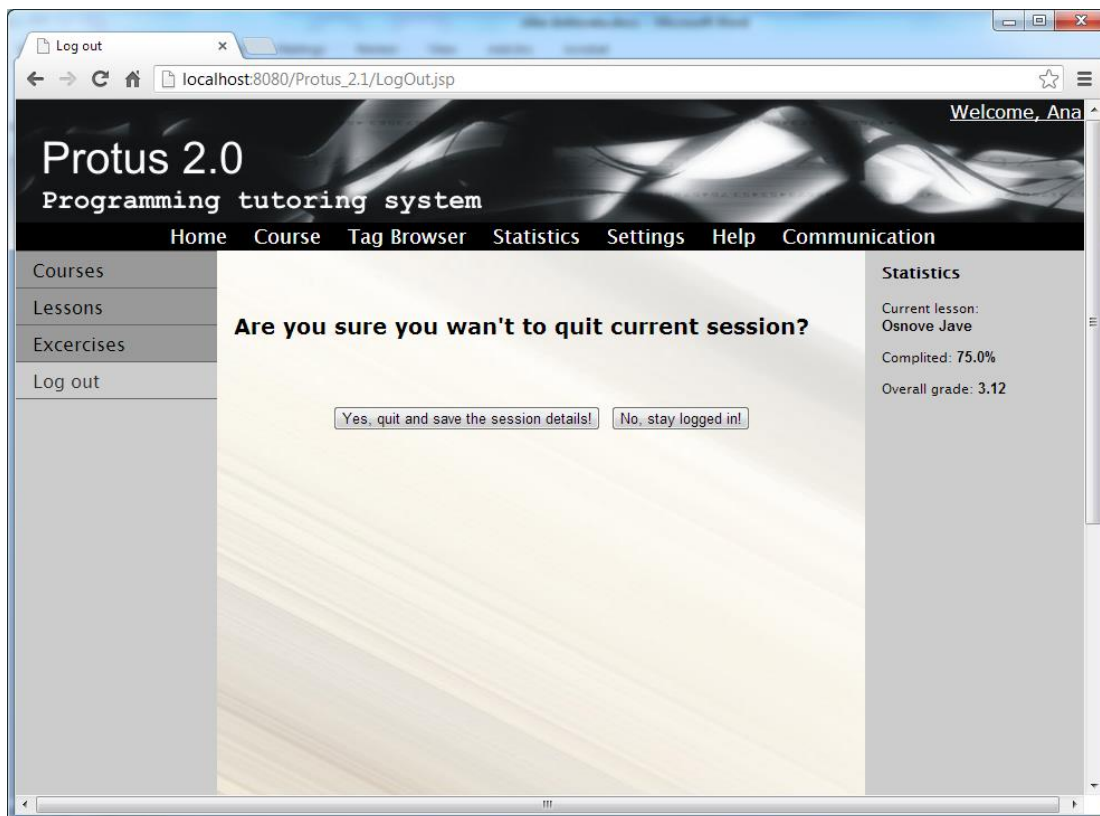
Slika 85. Pregled unetih tagova





Slika 86. Stranica sistema za podešavanje osnovnih opcija

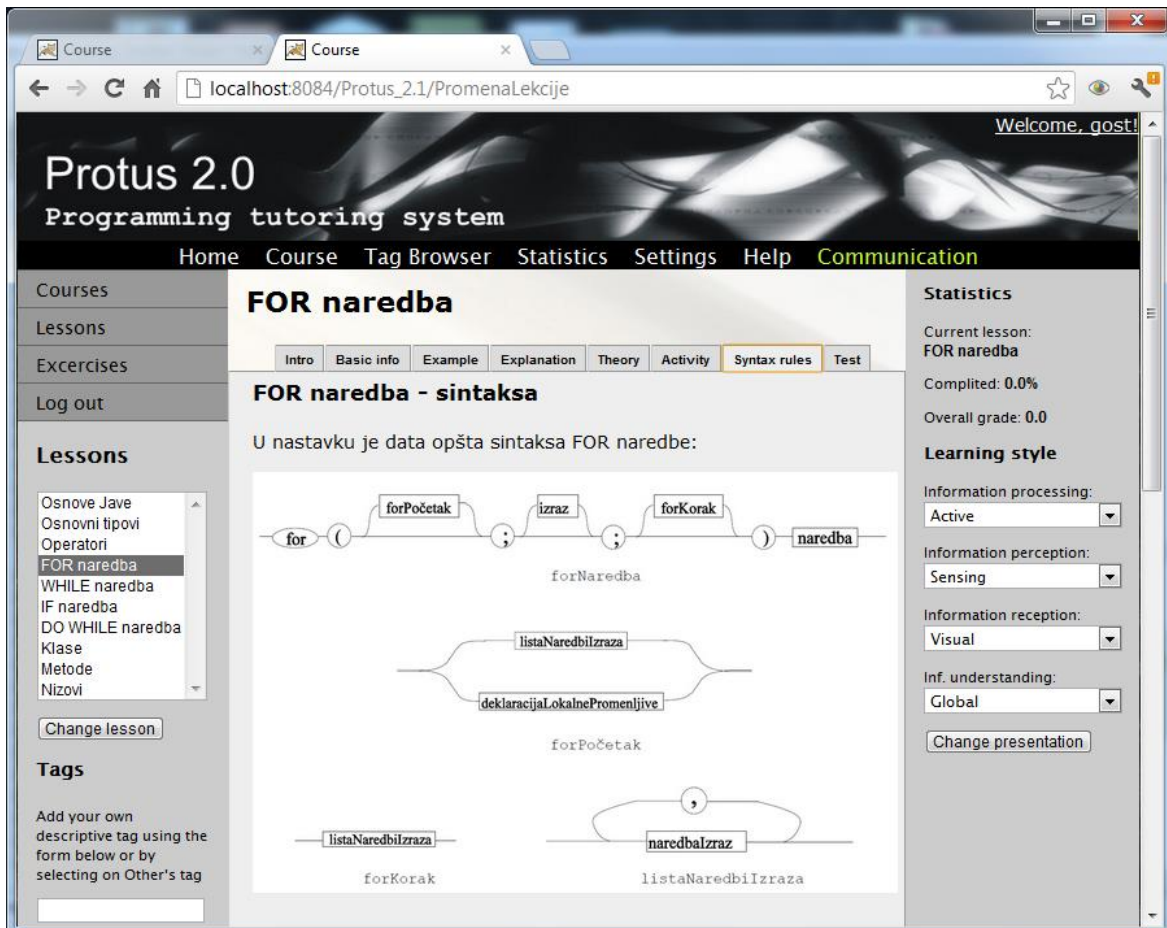
Opcija *Log out* zatvara trenutnu korisničku sesiju i ažurira ontologiju modela učenika sa podacima koji su o njemu formirani u toku sesije (slika 87).



Slika 87. Odjava učenika sistema

Na osnovu definisanih ontologija i pravila adaptacije, sistem Protus 2.0 personalizuje korisnički interfejs za svakog pojedinačnog učenika. Personalizacija uključuje prezentaciju preporučenog resursa i linkova kao rezultat pokretanja pravila adaptacije opisanih u poglavlju 6.3.

Slika 88 prikazuje korisnički interfejs sistema Protus 2.0 (za lekciju *for naredbe*), personalizovanog za učenika koji pripada kategorijama refleksivnog, intuitivnog, verbalnog i globalnog stila učenja u okviru domena procesiranja informacija, percepcije informacija, prijema informacija i razumevanja informacija.



Slika 88. Rezultati personalizacije u sistemu Protus 2.0

## **Poglavlje VIII**

### **DOPRINOSI DISERTACIJE I ZAVRŠNA RAZMATRANJA**

Disertacija prikazuje analizu implementacije različitih oblika personalizacije u tutorskom sistemu. Kao rezultat istraživanja predložen je model tutorskog sistema za realizaciju kurseva i nastavnog materijala iz različitih domena. Model je u potpunosti razvijen uz upotrebu tehnologija semantičkog veća i predstavlja osnovu za izgradnju veb aplikacije za elektronsko učenje. Predstavljani su osnovni koncepti arhitekture tutorskog sistema kao i relacije između njih. Kao potvrda valjanosti predloženog modela izvršeno je unapređenje postojećeg tutorskog sistema Protus.

Jedan od ciljeva upotrebe tehnologija semantičkog veća je rešavanje problema održavanja i ponovne upotrebe komponentata tutorskog sistema. Nastavni resursi u različitim sistemima za elektronsko učenje se definišu, strukturiraju i prikazuju u različitim oblicima. Da bi se omogućila ponovna upotreba resursa u različitim sistemima neohodno je definisati i prikazati resurse u standardizovanom obliku.

Prilikom realizacije sistema Protus 2.0 formalno je definisan opšti oblik koncepata i resursa za prikaz nastavnog materijala prilagođen različitim kursevima a različiti oblici personalizacije su predstavljeni precizno definisanim pravilima adaptacije.

Upotreba tehnologija semantičkog veća prilikom izrade sistema za elektronsko učenje najčešće se svodi na definisanje znanja iz domena. Model arhitekture predstavljen u disertaciji ne prikazuje samo znanje iz domena uz upotrebu ontologija, nego su ontologije upotrebljene prilikom izgradnje i ostalih komponenti sistema. Na taj način je svaka komponenta tutorskog sistema predstavljena odgovarajućom ontologijom.

Definisana arhitektura omogućuje:

- razvoj standardizovanog nastavnog materijala iz različitih domena,
- formalno definisanje procesa personalizacije sistema uz pomoć pravila adaptacije,
- jasno razdvajanje komponenti tutorskog sistema i njihovih funkcionalnosti,

- eksplicitnu komunikaciju između komponenti sistema,
- lako proširenje i adaptaciju komponenti sistema,
- lakše održavanje sistema,
- automatizaciju procesiranja/obrade informacija u tutorskom sistemu,
- integraciju sistema Protus 2.0 sa srodnim sistemima, razmenu ontologija i pravila adaptacije,
- olakšanu modifikaciju funkcionalnosti sistema.

Arhitektura sistema bazirana na tehnologijama semantičkog veba nudi bolju operativnost i ponovnu upotrebu sistemskih komponenti kao i mogućnosti proširenja sistema dodatnim programerskim kursevima kao i kursevima iz drugih oblasti.

Izgradnja arhitekture sistema obuhvatala je četiri osnovne faze (Vesin et al. 2011b):

- razvoj modela znanja prikazanog u ontologijama (definisane forme nastavnog materijala za prikaz koncepata i resursa u različitim tekstualnim ili grafičkim oblicima),
- prikaz znanja iz domena uz pomoć ontologija,
- uspostavljanje relacija između instanci klasa ontologija i definisanje pravila adaptacije uz pomoć SWRL jezika,
- konkretizacija znanja na osnovu šeme ontologija.

Prethodno navedene faze nude osnovu za izgradnju opšteg modela tutorskog sistema u formi ontologija. Ontologije su dopunjene pravilima adaptacije čija je uloga definisanje aktivnosti sistema u cilju personalizacije korisničkog interfejsa i nastavnog materijala tokom korisničkih sesija. U radu su prikazana SWRL pravila za personalizaciju sistema i definisanje i modifikaciju modela učenika.

Prikazane ontologije mogu poslužiti kao osnova za definiciju ontologija drugih adaptivnih sistema elektronskog učenja. Upotreba ontologija za modeliranje znanja u sistemu Protus 2.0 predstavlja ključni doprinos integraciji različitih informacija i nastavnog materijala koji se prikazuje učeniku, za unapređenje pretrage informacija i pre svega za definisanje i postavljanje upita nad znanjem sadržanim u ontologijama.

U sistemu Protus 2.0, ontologije se koriste za:

- **Modeliranje obrazovnih domena.** Ontologija za prikaz znanja iz domena definiše strukturu i ulogu svakog nastavnog resursa. Opšta forma resursa omogućuje obradu i prikaz znanja iz različitih domena.
- **Izgradnju, organizaciju i ažuriranje specifičnih obrazovnih resursa** (npr. nastavnog materijala, profila učenika, pitanja kroz nastavni materijal, itd). U sistemu su definisane specifične ontologije za svaku kategoriju obrazovnih resursa: ontologija za prikaz znanja iz domena, ontologija modela učenika sistema i ontologija nastavne strategije koje omogućuju skladištenje svih podataka relevantnih za nastavni proces.

Ove ontologije se takođe mogu koristiti za generisanje semantičkih pretraga, definisanje upita nad višestrukim repozitorijumima i za pronalaženje semantičkih relacija između pojedinih resursa i koncepata.

Kao konačan rezultat rada na disertaciji razvijen je potpuno funkcionalan tutorski sistem za elektronsko učenje koji učenicima nudi:



- mogućnosti pohađanja kurseva iz različitih domena adaptiranog svakom pojedinačnom učeniku,
- mogućnosti provere znanja učenika uz pomoć adekvatnih testova,
- potpuno funkcionalan kurs za učenje programskog jezika Java,
- mogućnost unosa novih kurseva i nastavnog materijala,
- opšti oblik i strukturu proširivih pravila adaptacije,
- skup pravila adaptacije iz opšteg domena koji se mogu prilagođavati specifičnim zahtevima različitih kurseva,
- ponovnu upotrebu komponenti sistema baziranih na tehnologijama semantičkog veba,
- mogućnosti primene tri osnovne kategorije personalizacije nastavnog materijala i korisničkog interfejsa (prilagođavanje korisničkog interfejsa individualnim stilovima učenja, generisanje preporuka nastavnog materijala, personalizacija prilagođavanjem linkova),
- mogućnosti adaptacije sistema specifičnim potrebama svakog učenika,
- mogućnosti dodavanja novih opcija personalizacije,
- formalizovan nastavni materijal.

Osnovni doprinos istraživanju prikazan u ovom radu je:

- Definisane opšteg modela tutorskog sistema. Modelirani su elementi tutorskog sistema za izvođenje kurseva iz različitih domena upotrebom tehnologija semantičkog veba.
- Definicija ontologija i pravila adaptacije u tutorskom sistemu, sa naglaskom na njihovoj razumljivosti i mogućnosti lakog ažuriranja i ponovne upotrebe. Svaka komponenta tutorskog sistema je modelirana odgovarajućom ontologijom, čime je omogućeno jasno razdvajanje komponenti tutorskih sistema (modula adaptacije, aplikacionog modula, modula domena, modela učenika i modula za praćenje korisničkih sesija) i eksplicitna komunikacija između njih. Pravila omogućavaju preciznu definiciju svih aktivnosti personalizacije koje se vrše u sistemu.
- Za svaku akciju personalizacije u sistemu na osnovu individualnih stilova učenja su definisane odgovarajuća pravila adaptacije. Stoga, Protus 2.0 sadrži pravila za prikazivanje opcija personalizacije za sva četiri domena stilova učenja.
- Definisane personalizovanog tutorskog sistema sa jasno izdvojenim komponentama. Sve akcije personalizacije su prikazane u formi SWRL pravila. Ova pravila adaptacije koriste podatke iz ontologija domena, zadataka, modela učenja i ontologije nastavne strategije u cilju izvršavanja personalizacije.
- Sve funkcionalnosti personalizacije u sistemu su predstavljene pravilima adaptacije. Na taj način, za modifikaciju opcija personalizacije u sistemu Protus 2.0 je dovoljno samo ažuriranje pravila adaptacije bez potrebe za menjanjem struktura nastavnog materijala i arhitekture sistema.

Sistem Protus 2.0 se sastoji od jasno definisanih i odvojenih komponenti koje se po potrebi mogu prilagođavati i dopunjavati. Moguća je i laka izmena opcija i mogućnosti personalizacije nastavnog materijala koji se nudi učenicima uz pomoć eksplicitno definisanih i dopunjivih pravila adaptacije.

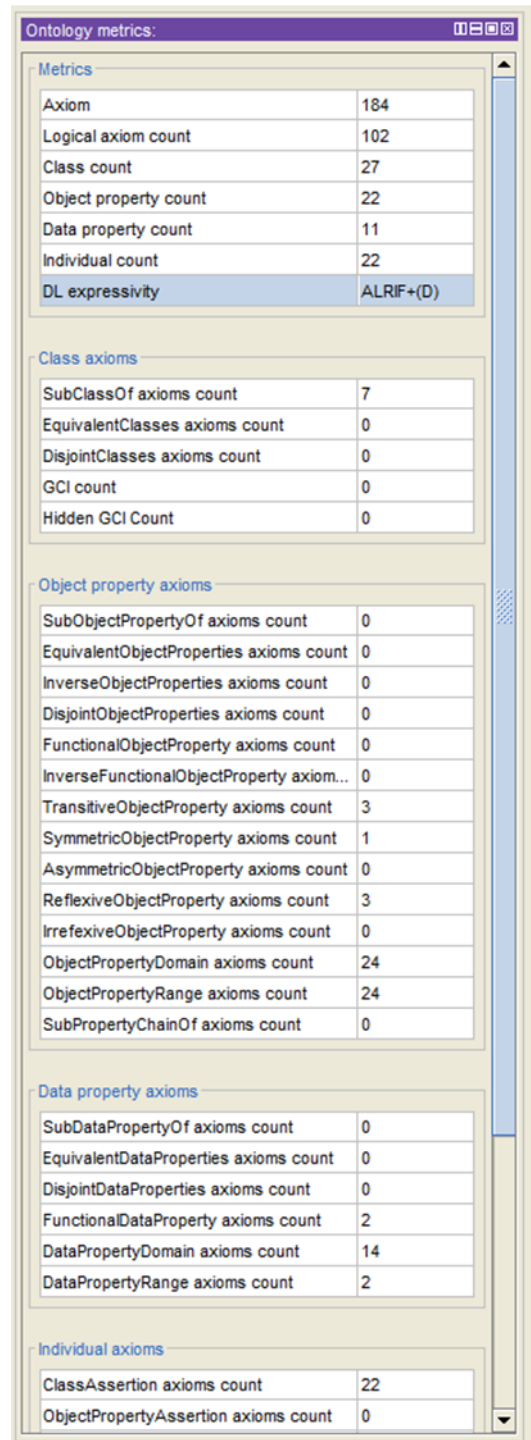
## 8.1 Metrike ontologija

Razvoj softvera uz pomoć tehnika semantičkog veća predstavlja izazov u smislu vremena i napora potrebnog prilikom kreiranja ontologija. Osnovni cilj je kreiranje ontologija koje će biti sadržajne, precizne i ponovo upotrebljive. Ontologije se kao i druga softverska rešenja moraju analizirati da bi se sprečilo da aplikacije koriste neprecizne, nekonzistentne ili netačne ontologije.

I u slučaju da ontologije ne uzrokuju nikakve probleme i da su formalno ispravne, korisnici/programeri moraju doneti odluku da li sadržaj ontologija odgovara zahtevima projekta na kojima su angažovani. Tačnije, korisnici/programeri moraju da odluče da li metapodaci i skup definisanih koncepata odgovaraju zahtevima koju su postavljeni pred njih. Alati za analizu ontologija mogu pomoći u proceni da li postojeće ontologije odgovaraju zahtevima korisnika. Sistemi Protégé i OntoQA predstavljaju najpopularnije alate za analizu ontologija (García-Peñalvo & Therón, 2010).

Metrike ontologija (eng. ontology metrics) predstavljaju važan pristup proceni kvaliteta ontologija. Iz ugla dizajnera ontologija, procenom kvaliteta ontologija moguće je prepoznati delove na kojima treba dodatno raditi ili uočiti delove koji bi mogli stvoriti problem. U literaturi postoje različiti predlozi procene: *koliko je dobra neka ontologija?*. Autori u (Bouillon et al., 2002) predlažu procenu kompletnosti, konzistentnosti i ispravnosti u smislu provere nedostatka nepravilnosti i grešaka. Drugi autori predlažu razvoj metodologija za analizu ontologija tokom celog procesa njihove implementacije (Gangemi et al., 2006). Naravno, mnogi autori tvrde da je jedino merilo kvaliteta ontologije analiza rada aplikacija koja ih koristi.

Postoje različite metode organizacije znanja u zavisnosti od potrebnog nivoa detalja i pravila u datom domenu. Stoga, ontologije se međusobno razlikuju ne samo po sadržaju već i po strukturi i implementaciji. Ontologije se mogu klasifikovati po detaljnosti koncepata i tipovima relacija između koncepata koje podržavaju (Seremeti & Kameas,



Ontology metrics:	
<b>Metrics</b>	
Axiom	184
Logical axiom count	102
Class count	27
Object property count	22
Data property count	11
Individual count	22
DL expressivity	ALRIF+(D)
<b>Class axioms</b>	
SubClassOf axioms count	7
EquivalentClasses axioms count	0
DisjointClasses axioms count	0
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
SubObjectPropertyOf axioms count	0
EquivalentObjectProperties axioms count	0
InverseObjectProperties axioms count	0
DisjointObjectProperties axioms count	0
FunctionalObjectProperty axioms count	0
InverseFunctionalObjectProperty axioms count	0
TransitiveObjectProperty axioms count	3
SymmetricObjectProperty axioms count	1
AsymmetricObjectProperty axioms count	0
ReflexiveObjectProperty axioms count	3
IrreflexiveObjectProperty axioms count	0
ObjectPropertyDomain axioms count	24
ObjectPropertyRange axioms count	24
SubPropertyChainOf axioms count	0
<b>Data property axioms</b>	
SubDataPropertyOf axioms count	0
EquivalentDataProperties axioms count	0
DisjointDataProperties axioms count	0
FunctionalDataProperty axioms count	2
DataPropertyDomain axioms count	14
DataPropertyRange axioms count	2
<b>Individual axioms</b>	
ClassAssertion axioms count	22
ObjectPropertyAssertion axioms count	0

Slika 89. Rezultati merenja ontologije sistema Protus 2.0

2010). Tako se razlikuju ontologije (sortirane od ontologija sa najmanjom ka ontologijama sa najvećom kompleksnošću strukture):

- terminološke ontologije (eng. *terminological ontologies*) koje samo definišu termine koji se koriste za prikaz znanja iz domena,
- informacione ontologije (eng. *information ontologies*) koje definišu strukturu koncepata uz definiciju njihovih atributa,
- ontologije za modeliranje znanja (eng. *knowledge modeling ontologies*) koje definišu konceptualizaciju znanja i imaju bogatu unutrašnju strukturu.

U zavisnosti od svoje opštosti, ontologije se mogu klasifikovati u četiri grupe (García-Peñalvo & Therón, 2010):

- ontologije aplikacije koje sadrže sve definicije potrebne za modeliranje znanja koje se koristi u okviru jedne aplikacije,
- ontologije domena koje se definišu u cilju konceptualizacije određenog domena,
- ontologije višeg nivoa kojima se definišu opštiji koncepti (koji se mogu koristiti u više različitih aplikacija) kao što su koncepti koji prikazuju stanja, događaje, akcije, itd.,
- ontologije reprezentacije koje služe za opis konceptualizacije sakrivene ispod formalizama prikaza znanja.

Uz pomoć alata za merenje ontologija integrisanog u sistem Protégé, može se doći do osnovnih podataka o veličini i kompleksnosti kreiranih ontologija u okviru sistema Protus 2.0 (slika 89). Radi se o ontologiji srednje veličine koja sadrži 27 klasa, 22 relacije definisane između tih klasa i 7 relacija nadklasa/podklasa. Iako se ne radi o velikoj ontologiji u njoj je sadržano svo znanje potrebno za formiranje funkcionalnog modela učenika i sadržajnog znanja iz domena.

Tabela 13 prikazuje poređenje ontologije sistema Protus 2.0 i postojećih ontologija sličnih tutorskih sistema. Prikazani su podaci o ontologiji sistema *Personal Reader* (Henze et al., 2004), ontologiji domena za učenje programiranja u programskog jeziku C, razvijenoj na Univerzitetu u Pittsburgh-u (Sosnovsky et al., 2007) i ontologiji sistema za prikaz nastavnog materijala iz teorije automatskog upravljanja (eng. *Control engineering*) prikazanoj u (Tong et al., 2010).

**Tabela 13. Usporedni prikaz ontologija**

	Personal Reader	Programski jezik C	CE ontologija	Protus 2.0
Broj definisanih klasa	490	575	21	27
Broj definisanih relacija	0	3	10	22
Broj relacija podklasa/nadklasa	0	2	2	7

Ontologija sistema *Personal Reader* se koristi samo u cilju prikaza postojećeg onlajn Java Oracle tutorijala. Ontologija domena programiranja u programskog jeziku C ima nešto opštiju strukturu ali i dalje prikazuje samo usko specijalizovan nastavni materijal. Ontologija sistema Protus 2.0 predstavlja ontologiju opšte namene tj. namenjena je kreiranju kurseva iz različitih domena.

Opštija struktura ontologije sistema Protus 2.0 je stoga neuporedivo manja od dve prethodno navedene ontologije, čije klase služe samo za prikaz sadržaja postojećih tutorijala.

U odnosu na ontologiju prikazanu u (Tong et al., 2010), ontologija sistema Protus je približno iste veličine ali je definisana sa više detalja i nudi bogatiju semantiku relacija između klasa.

Sadržajnost ontologije sistema Protus 2.0 proizilazi iz kompleksnosti njene strukture. Ontologija ne samo da definiše osnovne koncepte i njihovu strukturu, nego su detaljno definisani atributi koncepata kao i međusobne relacije između koncepata. Na taj način je detaljno definisana semantika svih elemenata sistema i pokriveni su složeni koncepti za prikaz različitih stanja, događaja ili akcija. Ontologije sistema Protus 2.0 omogućuju detaljnu konceptualizaciju ne samo domena za implementaciju programerskog kursa nego i domena tutorskog sistema opšte namene.

## **Poglavlje IX**

### **ZAKLJUČAK**

Elektronsko učenje predstavlja važan segment obrazovnih tehnologija. Ono predstavlja jedinstvenu priliku da se uči samostalno, nezavisno od vremena i mesta, da se stiče znanje bez prekida i prilagođeno pojedincu a na osnovu principa tradicionalnog obrazovanja. Danas, najpopularniji oblici elektronskog učenja su u formi sistema za učenje preko mreže ili uz upotrebu virtuelnih učionica i tutorskih sistema.

Sistemi za elektronsko učenje koriste različite tehnike sistema za generisanje preporuka u cilju izbora odgovarajućeg nastavnog materijala i aktivnosti učenika na osnovu njihovih potreba, znanja i stilova učenja. Sistemi za generisanje preporuka u sistemima za elektronsko učenje prilagođavaju nastavni materijal i korisnički interfejs specifičnim potrebama i zahtevima učenika.

U disertaciji je predstavljen pristup razvoju opšteg modela personalizovanog veb tutorskog sistema za pohađanje kurseva iz različitih domena. Na osnovu definisanog modela razvijen je sistem Protus 2.0. Sistem automatski prilagođava nastavni materijal i korisnički interfejs, zahtevima, navikama i nivou znanja svakog pojedinačnog učenika. Razlike između učenika se utvrđuju na osnovu trenutnog nivoa znanja, individualnog stila učenja, karakteristika, zahteva i ciljeva učenika. Sistem automatski usmerava aktivnosti učenika i generiše preporuke linkova, akcija i nastavnog materijala tokom procesa učenja.

U radu je prikazano kako se tehnologije semantičkog veba, ontologije i pravila adaptacije mogu koristiti za poboljšanje karakteristika postojećeg tutorskog sistema. Prikazana je arhitektura personalizovanog tutorskog sistema koji se u potpunosti oslanja na tehnologije i standarde semantičkog veba. Prikazane su ontologije koje odgovaraju komponentama tradicionalnih tutorskih sistema.

Arhitektura sistema u potpunosti podržava upotrebu tehnologija semantičkog veba za izgradnju osnovnih elemenata sistema i definisanje pravila personalizacije.

Ontologije će u potpunosti promeniti način na koji su sistemi konstruisani i organizovani. I dalje su velike baze znanja formirane bez mnogo mogućnosti deljenja informacija i

ponovne upotrebe. U budućnosti će razvoj inteligentnih tutorskih sistema biti olakšan velikom bibliotekom ontologija. Umesto da se razvoj takvih sistema pokreće iz početka, biće dovoljno da se potrebne komponente sistema izdvajaju iz postojećih biblioteka i repozitorijuma. Na taj način će se skratiti vreme razvoja a poboljšati robustnost i pouzdanost novoformiranih baza znanja i samih tutorskih sistema.

Eksplisitna konceptualizacija komponenti sistema u formi ontologija podstiče razmenu i ponovnu upotrebu znanja, komunikaciju i saradnju između komponenti sistema. Unapređena je upotreba ontologija za konstruisanje sistema koji zahtevaju eksplicitno strukturirano znanje. Takvi sistemi omogućuju učeniku pristup većem rasponu informacija i resursa. Prikazana arhitektura je modularna te omogućuje veću fleksibilnost i mogućnosti zamene pojedinih komponenti sve dok odgovaraju trenutnom interfejsu. Definisane ontologije mogu služiti kao početna baza znanja koja se može dalje proširivati i modifikovati u cilju definisanja domena različitih adaptivnih sistema.

Iako ontologije nude skup indirektnih tehnika rezonovanja na osnovu opisne logike na kojoj se zasnivaju (klasifikacija, identifikacija instanci klasa), ipak je neophodna implementacija pravila adaptacije da bi se definisala dodatna ograničenja koja moraju biti zadovoljena i da bi se omogućila automatska izmena baze znanja. Stoga se ontologijama pridodaju sistemi pravila adaptacije kojima se može generisati dodatno znanje iz već postojećeg. Sa druge strane i pravila adaptacije zahtevaju postojanje zajedničke taksonomije elemenata sistema u obliku ontologija kojom će se definisati koncepti i odnosi između njih. U radu je prikazan kompletan skup pravila adaptacije kojima se omogućuje rezonovanje nad instancama ontologija.

Osnovni uspeh ovog rada predstavlja sledeće:

- definisanje opšteg modela tutorskog sistema uz pomoć tehnologija semantičkog veba,
- definisanje odvojenih komponenti personalizovanog tutorskog sistema u formi obrazovnih ontologija,
- eksplicitan prikaz pravila adaptacije čime se olakšava razumevanje, ažuriranje i ponovna upotreba komponenti tutorskog sistema,
- implementacija različitih opcija personalizacije u tutorskom sistemu i
- prikaz mogućnosti upotrebe tehnologija semantičkog veba za izgradnju tutorskog sistema.

Ova arhitektura predstavlja dobru osnovu za dalje proširenje sistema i omogućuje detaljno modeliranje procesa personalizacije. Takođe, pravila adaptacije se mogu modifikovati u cilju postizanja specifičnih zahteva prilikom personalizacije sistema i modelovanja učenika.

Prikazana arhitektura tutorskog sistema sa elementima semantičkog veba omogućuje implementaciju i drugih programerskih kurseva ili kurseva iz drugih domena sa minimalnim izmenama u definisanoj arhitekturi. Izmene bi uključivale dodavanje novog nastavnog materijala i modifikovanje opcija za testiranje znanja učenika.

Buduća istraživanja će uključiti analizu uticaja greške prilikom inicijalne identifikacije stila učenja na dalji proces učenja i ponašanje sistema. Potrebno je unaprediti Protus 2.0 funkcionalnostima za praćenje ponašanja učenika i vođenje evidencije o broju učenika koji su menjali stil učenja i o vrstama samih promena.

## Literatura

- [1] Adomavicius G. & Tuzhilin A. (2005). *Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions*. IEEE Transactions on Knowledge and Data Engineering, 17(6), pp. 734–749.
- [2] Alsultanny Y. (2006). *e-Learning system overview based on semantic web*. The Electronic Journal of e-Learning 4 (2), pp. 111 – 118.
- [3] Aroyo L. & Mizoguchi R. (2003). *Process-aware authoring of web-based educational systems*. In Proceedings of the International Workshop on Semantic Web and Web-based Education SWWL, Velden, Austria, pp. 212-221.
- [4] Badica C. Budimac Z., Burkhard H. & Ivanović M. (2011). *Software agents: languages, tools, platforms*. Computer Science and Information Systems, 8 (2), pp. 255-298
- [5] Berners-lee T. (2000). *Semantic Web - XML2000*. W3C Website preuzeto 14. Septembra 2010 sa: <http://www.w3.org/2000/>
- [6] Berners-Lee T., Hendler J. & Lassila O. (2001). *The semantic web*. Scientific Am., pp. 34–43.
- [7] Breslin, J. G., Passant, A., & Vrandec, D. (2011). *Social semantic web*. In: Domingue, John; Fensel, Dieter; Hendler, James A. (Eds.), Handbook of Semantic Web Technologies Vol. 2, Springer, pp. 467-506.
- [8] Brusilovsky P. (2003). *A distributed architecture for adaptive and intelligent learning management systems*. In Proceedings of the AIED 2003 Workshop Towards Intelligent Learning Management Systems, Sydney, pp. 5-13.
- [9] Cardoso J. (2007). *The semantic web vision: where are we?*, IEEE Intelligent Systems, 22 (5), pp. 84-88.
- [10] Carmagnola F., Cena F., Gena C. & Torre I. (2005). *A semantic framework for adaptive web-based systems*. In Proceedings of SWAP 2005, the 2nd Italian semantic web workshop, Trento, Italy, doi:10.1.1.64.4246.
- [11] Chandrasekaran B., Josephson J. R. & Benjamins V. R., (1999). *What are ontologies, and why do we need them?*, IEEE Intelligent Systems, 4 (1), pp. 20-26
- [12] Chen C. M. (2008). *Ontology-based concept map for planning a personalized learning path*. Cybernetics and Intelligent Systems, pp. 1337-1342.
- [13] Chi Y. L. (2009). *Ontology-based curriculum content sequencing system with semantic rules*. Expert Systems with Applications, 36, pp. 7838–7847.
- [14] Coffield F., Moseley D., Hall E. & Ecclestone K. (2004). *Should we be using learning styles? What research has to say to practice*. Learning and Skills Research Centre, London, England
- [15] De Bra P. (2006). *Web based educational hypermedia*. In: Romero C. & Ventura S. (Eds.), Data mining in e-learning WIT press, Southampton, Boston, UK, pp. 3-17
- [16] De Bra P., Aroyo L. & Chepegin V. (2004). *The next big thing: adaptive web-based systems*. Journal of Digital Information 5, Article No. 247.
- [17] Dehors S. & Faron-Zucker C. (2006). *QBLS: A semantic web based learning system*. In Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 795-802.
- [18] Devedžić V. (2004a). *Education and the semantic web*, International Journal of Artificial Intelligence in Education 14, IOS Press, pp. 39-65.

- [19] Devedžić V. (2004b) *Web intelligence and artificial intelligence in education*. Educational Technology & Society, 7, 2004b, pp. 29- 39.
- [20] Devedžić V. (2006). *Semantic web and education*, Springer Science, New York
- [21] Diković L. J. (2009). *Applications GeoGebra into teaching some topics of mathematics at the college level*. Computer Science and Information Systems - ComSIS, 6(2), pp. 191–203.
- [22] Dolog P. & Nejdil W. (2007). *SemanticWeb technologies for the adaptive web*. The Adaptive Web, LNCS 4321, Springer-Verlag Berlin Heidelberg, pp. 697–719.
- [23] Dunn R., Dunn K. & Freeley M. E. (1984). *Practical applications of the research: responding to students' learning styles-step one*. Illinois State Research and Development Journal, 21(1), pp. 1–21.
- [24] Dutta B., (2006). *Semantic web based e-learning*, DRTC Conference on ICT for Digital Learning Environment, Bangalore, p.14
- [25] Emurian H. (2006). *A web-based tutor for java: evidence of meaningful learning*. International Journal of Distance Education Technologies, 4(2), pp. 10–30.
- [26] Farzan R. & Brusilovsky P. (2006). *Social navigation support in a course recommendation system*. In Proceedings of 4th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Dublin, pp. 91–100.
- [27] Felder R. M. & Silverman L. K. (1988). *Learning and teaching styles in engineering education*. Engineering Education, 78(7), pp. 674–681.
- [28] Felder R. M. & Soloman B. A. (1996). *Index of learning styles questionnaire*. preuzeto 17 Decembra 2009, sa: <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>.
- [29] Fensel D. & Musen M. A. (2001). *The semantic web: a brain for humankind*, IEEE Intelligent Systems, 16 (2), pp. 24-25.
- [30] Fernández-Breis J. T., Castellanos-Nieves D., Hernández-Franco J., Soler-Segovia C., Robles-Redondo M. C., González-Martínez R. et al. (2012). *A semantic platform for the management of the educative curriculum*. Expert Systems with Applications, 39(5), pp. 6011–6019.
- [31] Fertalj K., Hoić-Božić N. & Jerković H. (2010). *The integration of learning object repositories and learning management systems*. Computer Science and Information Systems, 7 (3), pp. 387-407.
- [32] Gangemi A., Catenacci C., Ciaramita M. & Lehmann J. (2006). *Modeling ontology evaluation and Validation*. The Semantic Web: Research and Application, LCNS 4011. pp. 140-154.
- [33] García-Peñalvo F. J. & Therón R. (2010). *A survey on ontology metrics*. Knowledge Management, Information Systems, E-Learning, and Sustainability Research Communications in Computer and Information Science, 111, pp. 22-27.
- [34] Gascueña J. M., Fernández-Caballero A. & González P. (2006). *Domain ontology for personalized e-learning in educational systems*. In The Sixth IEEE international conference on advanced learning technologies, pp. 456–458.
- [35] Gruber T. R. (1995). *Toward principles for the design of ontologies used for knowledge sharing*. Presented at the Padua workshop on Formal Ontology, March 1993, later published in International Journal of Human-Computer Studies, 43 (4-5), pp. 907-928.
- [36] Hammouda K. & Kamel M. (2006). *Data mining in e-learning*. In: Samuel Pierre (Ed.), E-Learning Networked environments and architectures: A knowledge processing Perspective Springer Book Series: Advanced Information and Knowledge Processing, pp. 374–402.



- [37] Hee Lee C., Hyun Seu J. & Evens M. W. (2002). *Building an ontology for CIRCSIMtutor*. In 13th Midwest AI and Cognitive Science Society Conference, Chicago, pp. 161–168.
- [38] Henze N. (2005). *Personal readers: personalized learning object readers for the semantic web*. In 12<sup>th</sup> International Conference on Artificial Intelligence in Education, AIED'05, Amsterdam, The Netherlands. pp. 274-281.
- [39] Henze N., Dolog P. & Nejd W. (2004). *Reasoning and ontologies for personalized e-learning in the semantic web*. Educational Technology & Society, 7 (4), pp. 82-97.
- [40] Holland J., Mitrovic A. & Martin B. (2009). *J-Latte: a constraint-based tutor for java*. In 17th International Conference on Computers in Education, Hong Kong, pp. 142–146.
- [41] Honey P. & Mumford A. (1982). *The manual of learning styles*. Maidenhead: Peter Honey.
- [42] Horrocks I., Parsia B., Patel-Schneider P. & Hendler J. (2005). *Semantic web architecture: stack or two towers?*, Lecture Notes in Computer Science, Volume 3703/2005, pp. 37-41.
- [43] Ivanović M., Pribela I., Vesin B. & Budimac Z. (2008). *Multifunctional environment for e-learning purposes*. Novi Sad Journal of Mathematics 38 (2), pp. 153–170.
- [44] Janssen J., Van den Berg B., Tattersall C., Hummel H. & Koper R. (2007). *Navigational support in lifelong learning: enhancing effectiveness through indirect social navigation*. Interactive Learning Environments, 15(2), pp. 127–136.
- [45] Jess (*Java Expert System Shell*), Preuzeto 7 Maja 2012. Sa: <http://jessrules.com/>.
- [46] Jia H., Wang M., Ran W., Yang S., Liao J. & Chiu D. (2011). *Design of a performance-oriented workplace e-learning system using ontology*. Expert Systems with Applications, 38(4), pp. 3372–3382.
- [47] Jovanović J., Rao R., Gašević D., Devedžić V. & Hatala M. (2007). *Ontological framework for educational feedback*. In Proceedings of the SWEL Workshop of Ontologies and Semantic Web Services for IES, AIED pp. 54-64.
- [48] Kelly D. & Tangney B. (2004). *Predicting learning characteristics in a multiple intelligence based tutoring system*. Berlin: Springer-Verlag, pp. 9–30.
- [49] Kim E. & Schniederjans M., (2004). *The role of personality in Web-based distance education courses*, Communications of the ACM, 47 (3), pp. 95-98.
- [50] Klačnja-Milićević A., Vesin B., Ivanović M. & Budimac Z. (2011a). *Integration of recommendations and adaptive hypermedia into Java tutoring system*. Computer Science and Information Systems – ComSIS, 8 (1), pp. 211-224.
- [51] Klačnja-Milićević A., Vesin B., Ivanović M. & Budimac Z. (2011b). *E-learning personalization based on hybrid recommendation strategy and learning style identification*. Computers & Education, 56, pp. 885–899.
- [52] Kolb D. (1984). *Individuality in learning and the concept of learning styles*. In Experiential learning Englewood Cliffs, New Jersey: Prentice-Hall. pp. 61–98.
- [53] Koper E. J. R. (2005). *Increasing learner retention in a simulated learning network using indirect social interaction*. Journal of Artificial Societies and Social Simulation, 8(2), pp. 18–27.
- [54] Kristofic A. (2005). *Recommender system for adaptive hypermedia Applications*. In Proceeding of Informatics and Information Technology Student Research Conference, Bratislava, pp. 229–234.

- [55] Lee M.C., Yen Ye D. & Wang T.I. (2005). *Java learning object ontology*. In Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies, pp. 538-542.
- [56] Linden G., Smith B. & York J. (2003). *Amazon.com recommendations: Item-to-item collaborative filtering*. IEEE Internet Computing, 7(1), pp. 76-80.
- [57] Merino P. J. M. & Kloos C. D. (2008). *An architecture for combining semantic web techniques with intelligent tutoring systems*. Springer-Verlag Berlin Heidelberg, Volume 5091/2008, pp. 540–550.
- [58] Mizoguchi R. & Bourdeau J. (2000). *Using ontological engineering to overcome AI-ED problems*. International Journal of Artificial Intelligence in Education, 11 (2), pp. 107-121.
- [59] Mizoguchi R., Hayashi Y. & Bourdeau J. (2007). *Inside theory-aware and standards-compliant authoring system*. SWEL Workshop of Ontologies and Semantic Web Services for IES, AIED, pp. 1-18.
- [60] Mustapaşa, O., Karahoca, D., Karahoca, A., Yücel, A., & Uzunboylu, H. (2010). Implementation of semantic web mining on e-learning. *Procedia - Social and Behavioral Sciences*. 2(2), pp. 5820–5823.
- [61] OWL (*Ontology Web Language*): preuzeto 25. oktobra 2013. sa: <http://www.w3.org/2004/OWL/>.
- [62] Padayachee I. (2000). *Intelligent Tutoring Systems: Architecture and Characteristics*, Information Systems & Technology, p. 8
- [63] Papataxiarhis V., Tsetsos V., Karali I., Stamatopoulos P. & Hadjiefthymiades S. (2010). *Developing rule-based applications for the web: Methodologies and tools*. Web Technologies: Concepts, Methodologies, Tools, and Applications, 4, pp. 456–477.
- [64] Parvez S. M. & Blank G. D. (2008). *Individualizing tutoring with learning style based feedback*. Intelligent Tutoring Systems 9th International Conference, Montreal, Springer, pp. 291–301.
- [65] Pashler H., McDaniel M., Rohrer D. & Bjork R. (2009). *Learning styles: concepts and evidence*. Psychological Science in the Public Interest, 9(3), pp. 105–119.
- [66] Pask G. (1976). *Styles and strategies of learning*. British Journal of Educational Psychology, 46, pp. 128–148.
- [67] Peña C.I., Marzo J.L. & de la Rosa J.L. (2002). *Intelligent agents in a teaching and learning environment on the web*. In: IEEE International Conference on Advanced Learning Technologies, pp. 21–27.
- [68] Popescu E., Badica C. & Trigano P. (2007). *Rules for learner modeling and adaptation provisioning in an educational hypermedia system*. In Ninth international symposium on symbolic and numeric algorithms for scientific computing, pp. 492–499.
- [69] Pritchard, A. (2013). *Ways of learning: learning theories and learning styles in the classroom*. Routledge, New York, Third Edition
- [70] Protégé: preuzeto 25. oktobra 2013. sa: <http://protege.stanford.edu/>.
- [71] RDF (*Resource Description Framework*): preuzeto 25. oktobra 2013. sa: <http://www.w3.org/RDF/>
- [72] Resnick P., Lacovou N., Suchak M., Bergstrom P. & Riedl J. (1994). *GroupLens: an open architecture for collaborative filtering of netnews*. In Proceedings of ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, pp. 175–186.

- [73] Romero C., Ventura S., Delgado J. A. & De Bra P. (2007). *Personalized links recommendation based on data mining in adaptive educational hypermedia systems*, Creating New Learning Experiences on a Global Scale, pp. 292-306.
- [74] Romero C. & Ventura S. (2006). *Preface*. In: Romero C. & Ventura S. (Eds.), *Data mining in e-learning* WIT press, Southampton, Boston, UK, pp. 3-19.
- [75] Romero C., Ventura S. & De Bra P. (2004). *Knowledge discovery with genetic programming for providing feedback to courseware*. *User Modeling and User-Adapted Interaction*, 14(5), pp. 425–464.
- [76] Ruttenbur B. & Spinkler G. (2000). *eLearning the Engine of the Knowledge Economy*, Morgan Keegen & co, inc.
- [77] Schafer J. B., Konstan J. & Riedl J. (1999). *Recommender systems in e-commerce*. In *Proceedings of 1st ACM conference on Electronic commerce*, Denver, Colorado, pp. 158-166.
- [78] Schafer J. B. (2005). *The application of data-mining to recommender systems*, *Encyclopedia of data warehousing and mining*, Hershey, PA. Idea Group, pp. 44–48.
- [79] SCORM 2004 3rd Edition - Overview, *Advanced Distributed Learning*. Preuzeto 16. Novembra 2006 sa: <http://www.adlnet.gov>.
- [80] Seremeti L., & Kameas A. (2010). *Tools for ontology engineering and management*. *Theory and Applications of Ontology: Computer Applications*, pp. 131-154.
- [81] Shah, N. K. (2012). *E-learning and semantic web*. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 2, 113–116.
- [82] Sheth A., Ramakrishnan C. & Thomas C. (2005). *Semantics for the Semantic Web: The Implicit, the Formal and the Powerful*, *International Journal on Semantic Web & Information Systems*, pp. 1-18.
- [83] Sicilia M. A., Lytras M. D., Sances-Alonso S., Garcia-Barriocanal E. & Zapata-Ros M. (2010). *Modeling instructional-design theories with ontologies: Using methods to check, generate and search learning designs*. *Computers in Human Behavior*, vol. 27, pp. 1389-1398.
- [84] Sosnovsky S., Dolog P., Henze N., Brusilovsky, P., & Nejdl W. (2007). *Translation of overlay models of student knowledge for relative domains based on domain ontology mapping*. *Frontiers in Artificial Intelligence and Applications*, 158, paper no. 289.
- [85] Sosnovsky S., Mitrovic A., Lee D.H., Brusilovsky P., Yudelso M. & Brusilovsky V. (2008). *Towards integration of adaptive educational systems: mapping domain models to ontologies*. In *Proceedings of the 6th International Workshop on Ontologies and Semantic Web for E-Learning at ITS*, Montreal, Canada, pp. 60-64.
- [86] Swartout W. & Tate A. (1999). *Ontologies*. *Intelligent Systems and their Applications*, IEEE, 14(1), pp. 18-19.
- [87] SWRL A Semantic Web Rule Language Combining OWL and RuleML. Preuzeto 18. septembra 2012. Sa: <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>
- [88] Sykes E.R. & Franek F. (2003). *An intelligent tutoring system prototype for learning to program java*. In: *The third IEEE International Conference on Advanced Learning Technologies*, Athens, Greece, pp. 485–492.
- [89] Strohmaier, M., Walk, S., Pöschko, J., Lamprecht, D., Tudorache, T., Nyulas, C., & Noy, N. F. (2013). *How ontologies are made: Studying the hidden social dynamics behind collaborative*

*ontology engineering projects*. Web Semantics: Science, Services and Agents on the World Wide Web, 20, pp. 18-34.

- [90] Šimić G. (2004). *The multi-courses tutoring system design*. Computer Science and Information Systems – ComSIS, 1(1), pp. 141–155.
- [91] Tang T. Y. & McCalla G. (2005). *Smart recommendation for an evolving e-learning system: architecture and experiment*. International Journal on e-Learning, 4(1), pp. 105–129.
- [92] Tong M. W., Liu Q. T., & Liu X. N. (2010). *A service context model based on ontology for content adaptation in E-learning*. In Frontiers in Education Conference (FIE), 2010 IEEE, pp. S1D-1
- [93] Vesin B. & Ivanović M. (2004). *Modern educational tools*. Proceedings of PRIM2004, 16th Conference on Applied Mathematics, Budva, Montenegro. 2004.
- [94] Vesin B., Ivanović M., Budimac Z. & Pribela I. (2008). *MILE – multifunctional integrated learning environment*. In IADIS Multi Conference on Computer Science and Information Systems MCCSIS'2008, Amsterdam, Netherlands, pp. 104–108.
- [95] Vesin B., Ivanović M. & Budimac Z. (2009). *Learning management system for programming in java*. Annales Universitatis Scientiarum De Rolando Eötvös Nominatae, Sectio Computatorica, 31, pp. 75–92.
- [96] Vesin B. Ivanović M., Klašnja-Milićević A. & Budimac Z. (2011a). *Rule-based reasoning for building learner model in programming tutoring system*, In: H. Leung et al. (Eds.): ICWL 2011, LNCS 7048, Springer, Heidelberg, pp. 154--163.
- [97] Vesin B., Ivanović M., Klašnja-Milićević A. & Budimac Z. (2011b) *Rule-based reasoning for altering pattern navigation in programming tutoring system*, 15th international conference on System theory, Control and Computing, October 14-16, Sinaia, Romania, pp. 644-649.
- [98] Vesin B., Klašnja-Milićević A., Ivanović M. & Budimac Z. (2012a). *Applying recommender systems and adaptive hypermedia for e-learning personalization*, Computing and Informatics, 32(3), pp. 629-659.
- [99] Vesin B., Ivanović M., Klašnja-Milićević A. & Budimac Z. (2012b). *Protus 2.0: ontology-based semantic recommendation in programming tutoring system*, Experts systems with application Vol 39, pp. 12229–12246.
- [100] Vesin B., Ivanović M., Klašnja-Milićević A. & Budimac Z. (2013). *Applied recommendation supported with the ontology-based architecture in java tutoring system*, Computer Science and Information Systems – ComSIS, 10(1), pp. 237-261.
- [101] Vouk M., Bitzer D. & Klevans R. (1999). *Workflow and end-user quality of service issues in web-based education*, IEEE Transactions on Knowledge and Data Engineering, pp. 673 – 687.
- [102] Wand, Y., Storey, V. C., & Weber, R. (1999). *An ontological analysis of the relationship construct in conceptual modeling*. ACM Transactions on Database Systems. 24(4), pp. 494-528.
- [103] Wang H.C., Li T.Y. & Chang C.Y. (2004). *A web based tutoring system with styles matching strategy for learning spatial geometry*. In: International Computer Symposium, Taipei, Taiwan, pp. 226–233.
- [104] Wolf C. (2003). *iWeaver: Towards 'Learning Style'-based E-learning in Computer Science Education*, Conferences in Research and Practice in Information Technology Vol. 20, Adelaide, Australia: Australian Computer Society, pp. 273-279.

- [105] Wolf C. (2007). *Construction of an Adaptive E-learning Environment to Address Learning Styles and an Investigation of the Effect of Media Choice*, PhD Thesis, RMIT University, Melbourne
- [106] Zaïane O.R. (2002). *Building a recommender agent for e-learning systems*. In proceedings of The International Conference on Computers in Education, ICCE'02, pp. 55–59.

## **Sažetak**

Semantički veb je nova generacija veba u kojem se pokušava da se informacije predstave na takav način da ih računari mogu koristiti ne samo za prikazivanje nego i za automatizaciju, integraciju i ponovnu upotrebu između raznih aplikacija. Cilj disertacije je implementacija i predstavljanje svih elemenata tutorskog sistema za realizaciju kurseva iz različitih domena pomoću tehnologija semantičkog veba. Ovaj proces obuhvata kreiranje osnovnih gradivnih ontologija kao i pravila za izvođenje konkretnih akcija kojim se postiže personalizacija nastavnog materijala.

Predmet istraživanja disertacije obuhvata realizaciju opšteg modela tutorskog sistema za elektronsko učenje iz različitih domena primenom tehnologija semantičkog veba i primena tog modela za izgradnju tutorskog sistema za učenje programskog jezika Java sa elementima personalizacije.

## **Abstract**

Semantic web is a next generation of web that is trying to present information in such a way that they can be used by computers, for display, automation, integration and reuse among different applications. The aim of the dissertation is the implementation and presentation of all elements of the tutoring system for presentation of courses from various domains using semantic web technologies. This process includes the creation of the fundamental building blocks of ontologies and rules for carrying out the actions for adaptation of teaching materials.

The subject of the dissertation includes the implementation of a conceptual model of tutoring system for e-learning in different domains using semantic web technologies and application of that model in a design of a tutoring system for learning the Java programming language basics with personalization.

## **Prilozi**

## Prilog 1. Atributi ontologija modela tutorskog sistema

Tabela 1.1. Atributi ontologije znanja iz domena

<b>Concept</b>		
hasId[type:int]	Identifikacioni broj koncepta	Datatype property
hasame[type:string]	Naziv koncepta	Datatype property
hasPrerequisite	Koncept čije savladavanje je preduslov prelaska na trenutni	Object property
subConceptOf	nadkoncept	Object property
hasResource	Dodeljeni resurs	Object property
<b>Resurs</b>		
hasId[type:int]	Identifikacioni broj resursa	Datatype property
hasName[type:string]	Naziv resursa	Datatype property
isResourceFor	Koncept kojem pripada	Object property
supports[type: string]	Uloga resursa	Datatype property
isVisited[type: string]	Da li je resurs posećen	Datatype property
isRecommended[type: string]	Da li je resurs preporučen	Datatype property
hasFileType[type: string]	Tip fajla	Datatype property
hasRole[type: string]	Uloga resursa	Datatype property
hasFigure[type: string]	Link ka jpg fajlu	Datatype property
hasPrerequisite	Resurs čije savladavanje je preduslov prelaska na trenutni	Object property
isTypeOf	Tip resursa	Object property
hasRole	Uloga resursa	Object property

Tabela 1.2. Atributi ontologije zadataka

<b>ResourceType</b>		
hasId[type:int]	Identifikacioni broj tipa resursa	Datatype property
hasName[type:string]	Naziv tipa resursa	Datatype property
<b>ResourceRole</b>		
hasId[type:int]	Identifikacioni broj uloge resursa	Datatype property
hasName[type:string]	Naziv uloge resursa	Datatype property
hasType	Tip uloge resursa	Object property
subRoleOf	Nad tip uloge resursa	Object property
<b>ResourceRoleType</b>		
hasId[type:int]	Identifikacioni broj tipa uloge resursa	Datatype property
hasName[type:string]	Naziv tipa uloge resursa	Datatype property



Tabela 1.3. Atributi ontologije modela učenika

<b>User</b>		
id [type:int]	Identifikacioni broj korisnika	Datatype property
<b>Teacher</b>		
id [type:int]	Identifikacioni broj predavača	Datatype property
isTypeOf	Nadklasa	Object property
<b>Learner</b>		
id [type:int]	Identifikacioni broj učenika	Datatype property
isTypeOf	Nadklasa	Object property
hasPerformance	Rezultati rada učenika	Object property
hasInfo	Informacije o učeniku	Object property
hasLearningStyle	Stil učenja učenika	Object property
hasInteraction	Sprovedene interakcije	Object property
<b>PersonallInfo</b>		
learner[type:int]	Identifikacioni broj učenika	Datatype property
name[type:string]	Ime učenika	Datatype property
lastname[type:string]	Prezime učenika	Datatype property
gender[type:string]	Pol učenika	Datatype property
address[type:string]	Adresa učenika	Datatype property
birthDate[type:dateTimeStamp]	Datum rođenja	Datatype property
birthPlace[type:string]	Mesto rođenja	Datatype property
previousKnowledge[type:string]	Prethodno znanje	Datatype property
affiliation[type:string]	Zanimanje	Datatype property
<b>Performance</b>		
learner[type:int]	Identifikacioni broj učenika	Datatype property
course[type:int]	Trenutni kurs	Datatype property
percentage[type:double]	Procenat savladanosti kursa	Datatype property
hasGrade[type:float]	Trenutna ocena	Datatype property
avgGrade[type:double]	Prosečna ocena	Datatype property
lesson[type:int]	Poslednja savladana lekcija	Datatype property
curLesson[type:int]	Trenutna lekcija	Datatype property
resurs[type:int]	Trenutni resurs	Datatype property
generates	Uslovi koje generiše	Object property
<b>LearningStyle</b>		
learner[type:int]	Identifikacioni broj učenika	Datatype property
isCategoryOf[type:string]	Naziv kategorije	Datatype property
generates	Uslovi koje generiše	Object property
hasCategory	Kategorija stila učenika	Object property
<b>Interaction</b>		
hasId[type:int]	Identifikacioni broj interakcije	Datatype property
partOf	Veza sa odgovarajućom sesijom	Object property
hasType	Tip interakcije	Object property
conceptUsed	Korišćeni koncept	Object property
whoInteracted	Učenik koji vrši interakciju	Object property
hasResult	Dobijeni rezultat	Object property
<b>Session</b>		
hasId[type:int]	Identifikacioni broj sesije	Datatype property
began[type:dateTimeStamp]	Vreme početka sesije	Datatype property

ended[type:dateTimeStamp]	Vreme završetka sesije	Datatype property
<b>InteractionType</b>		
hasId[type:int]	Identifikacioni broj tipa interakcije	Datatype property
hasName[type:string]	Naziv tipa interakcije	Datatype property

Tabela 1.4. Atributi ontologije nastavne strategije

<b>Condition</b>		
hasId[type:int]	Identifikacioni broj generisanog uslova	Datatype property
hasName[type:string]	Naziv uslova	Datatype property
generatedBy	Stil učenja aktivnog učenika	Object property
hasLearningStyleCategory	Kategorija stila učenja	Object property
hasLearningStyleDomain	Dimenzija stila učenja	Object property
generates	Tip personalizacije koju generiše	Object property
<b>Personalisation</b>		
hasId[type:int]	Identifikacioni broj personalizacije	Datatype property
basedOn	Uslov na osnovu koga je pokrenuta	Object property
determines	Odluka koju sprovodi	Object property
<b>BehaviourPattern</b>		
hasId[type:int]	Identifikacioni broj šablona	Datatype property
hasRate[type:double]	Rejting navigacione sekvence	Datatype property
isTypeOf	Tip sekvence kojoj pripada	Object property
partOf	Sesija kojoj pripada	Object property
generate	Personalizacija koju generiše	Object property
<b>NavigationSequence</b>		
hasId[type:int]	Identifikacioni broj sekvence	Datatype property
consistOf	Resurs koji mu pripada	Object property
Num [type:int]	Redni broj resursa u sekvenci	Datatype property
<b>CurrentGoal</b>		
hasId[type:int]	Identifikacioni broj trenutnog cilja učenja	Datatype property
isTypeOf	Tip cilja učenja	Object property
<b>TeachingGoals</b>		
hasId[type:int]	Identifikacioni broj tipa cilja učenja	Datatype property
hasName[type:string]	Naziv tipa cilja učenja	Datatype property
<b>Decision</b>		
hasId[type:int]	Identifikacioni broj odluke	Datatype property
hasCode[type:string]	Kod odluke	Datatype property
hasDescription[type:string]	Opis odluke	Datatype property

Tabela 1.5. Atributi ontologije korisničkog interfejsa

<b>Presentation</b>		
hasId[type:int]	Identifikacioni broj prezentacije	Datatype property
hasDescription[type:string]	Opis prezentacije	Datatype property
creates	Sekvenca koju kreira	Object property
implements	Odluka koja uslovljava prezentaciju	Object property

## Prilog 2. Izvorni kod implementiranih ontologija

```
<?xml version="1.0"?>

<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.protus.com/ontologies/domain.owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  ontologyIRI="http://www.protus.com/ontologies/domain.owl">
  <Prefix name="" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" />
  <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace" />
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Annotation>
    <AnnotationProperty abbreviatedIRI="rdfs:comment" />
    <Literal datatypeIRI="&rdf;PlainLiteral">Domain
ontology</Literal>
  </Annotation>
  <Declaration>
    <Class IRI="#Active/Reflective" />
  </Declaration>
  <Declaration>
    <Class IRI="#BehaviourPattern" />
  </Declaration>
  <Declaration>
    <Class IRI="#Concept" />
  </Declaration>
  <Declaration>
    <Class IRI="#Condition" />
  </Declaration>
  <Declaration>
    <Class IRI="#CurrentGoal" />
  </Declaration>
  <Declaration>
    <Class IRI="#Decision" />
  </Declaration>
  <Declaration>
    <Class IRI="#Interaction" />
  </Declaration>
  <Declaration>
    <Class IRI="#InteractionType" />
  </Declaration>
  <Declaration>
    <Class IRI="#Learner" />
  </Declaration>
  <Declaration>
    <Class IRI="#LearningStyle" />
  </Declaration>
  </Ontology>
```

```
</Declaration>
<Declaration>
  <Class IRI="#LearningStyleCategory"/>
</Declaration>
<Declaration>
  <Class IRI="#NavigationSequence"/>
</Declaration>
<Declaration>
  <Class IRI="#Performance"/>
</Declaration>
<Declaration>
  <Class IRI="#PersonalInfo"/>
</Declaration>
<Declaration>
  <Class IRI="#Personalization"/>
</Declaration>
<Declaration>
  <Class IRI="#Presentation"/>
</Declaration>
<Declaration>
  <Class IRI="#ResouceRoleType"/>
</Declaration>
<Declaration>
  <Class IRI="#Resource"/>
</Declaration>
<Declaration>
  <Class IRI="#ResourceRole"/>
</Declaration>
<Declaration>
  <Class IRI="#ResourceType"/>
</Declaration>
<Declaration>
  <Class IRI="#Sensing/Intuitive"/>
</Declaration>
<Declaration>
  <Class IRI="#Sequential/Global"/>
</Declaration>
<Declaration>
  <Class IRI="#Session"/>
</Declaration>
<Declaration>
  <Class IRI="#Teacher"/>
</Declaration>
<Declaration>
  <Class IRI="#TeachingGoals"/>
</Declaration>
<Declaration>
  <Class IRI="#User"/>
</Declaration>
<Declaration>
  <Class IRI="#Visual/Verbal"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#basedOn"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#conceptUsed"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#consistOf"/>
</Declaration>
```

```
<Declaration>
  <ObjectProperty IRI="#creates"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#determines"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#generates"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasCategory"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasInfo"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasInteraction"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasLearningStyle"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasPerformance"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasPrerequisite"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasResource"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasResult"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasRole"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasType"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#implements"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#isTypeOf"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#partOf"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#subConceptTypeOf"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#subRoleOf"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#supports"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasFileType"/>
</Declaration>
<Declaration>
```

```
    <DataProperty IRI="#hasGrade"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#hasId"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#hasLearningStyleCategory"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#hasLearningStyleDomain"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#hasName"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#hasRate"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#hasRole"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#isRecommended"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#isVisited"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#supports"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#BasicInfo"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Classes"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#CourseMaterial"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#DoWhileStatement"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#DomainResource"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Exam"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#ExaminationMaterial"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Example"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#ExecutionControl"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Exercise"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#ForStatement"/>
```

```

</Declaration>
<Declaration>
  <NamedIndividual IRI="#Goals"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Introduction"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#JavaProgrammingCourse"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#LoopStatements"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Methods"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Operators"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Syntax"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#SyntaxRule"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Task"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Theory"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#WhileStatement"/>
</Declaration>
<Declaration>
  <AnnotationProperty IRI="#birthDate"/>
</Declaration>
<Declaration>
  <AnnotationProperty IRI="#lastname"/>
</Declaration>
<Declaration>
  <AnnotationProperty IRI="#name"/>
</Declaration>
<Declaration>
  <AnnotationProperty IRI="#password"/>
</Declaration>
<Declaration>
  <AnnotationProperty IRI="#username"/>
</Declaration>
<SubClassOf>
  <Class IRI="#Active/Reflective"/>
  <Class IRI="#LearningStyleCategory"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Learner"/>
  <Class IRI="#User"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Sensing/Intuitive"/>
  <Class IRI="#LearningStyleCategory"/>
</SubClassOf>

```

```

<SubClassOf>
  <Class IRI="#Sequential/Global"/>
  <Class IRI="#LearningStyleCategory"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Teacher"/>
  <Class IRI="#User"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#TeachingGoals"/>
  <Class IRI="#CurrentGoal"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Visual/Verbal"/>
  <Class IRI="#LearningStyleCategory"/>
</SubClassOf>
<ClassAssertion>
  <Class IRI="#ResourceType"/>
  <NamedIndividual IRI="#BasicInfo"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Concept"/>
  <NamedIndividual IRI="#Classes"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ResourceType"/>
  <NamedIndividual IRI="#CourseMaterial"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Concept"/>
  <NamedIndividual IRI="#DoWhileStatement"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ResourceType"/>
  <NamedIndividual IRI="#DomainResource"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ResourceType"/>
  <NamedIndividual IRI="#Exam"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ResourceType"/>
  <NamedIndividual IRI="#ExaminationMaterial"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ResourceType"/>
  <NamedIndividual IRI="#Example"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Concept"/>
  <NamedIndividual IRI="#ExecutionControl"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#ResourceType"/>
  <NamedIndividual IRI="#Exercise"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Concept"/>
  <NamedIndividual IRI="#ForStatement"/>
</ClassAssertion>
<ClassAssertion>

```



```

    <Class IRI="#ResourceType"/>
    <NamedIndividual IRI="#Goals"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#ResourceType"/>
    <NamedIndividual IRI="#Introduction"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Concept"/>
    <NamedIndividual IRI="#JavaProgrammingCourse"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Concept"/>
    <NamedIndividual IRI="#LoopStatements"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Concept"/>
    <NamedIndividual IRI="#Methods"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Concept"/>
    <NamedIndividual IRI="#Operators"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Concept"/>
    <NamedIndividual IRI="#Syntax"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#ResourceType"/>
    <NamedIndividual IRI="#SyntaxRule"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#ResourceType"/>
    <NamedIndividual IRI="#Task"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#ResourceType"/>
    <NamedIndividual IRI="#Theory"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Concept"/>
    <NamedIndividual IRI="#WhileStatement"/>
</ClassAssertion>
<SymmetricObjectProperty>
    <ObjectProperty IRI="#hasType"/>
</SymmetricObjectProperty>
<TransitiveObjectProperty>
    <ObjectProperty IRI="#hasPrerequisite"/>
</TransitiveObjectProperty>
<TransitiveObjectProperty>
    <ObjectProperty IRI="#subConceptTypeOf"/>
</TransitiveObjectProperty>
<TransitiveObjectProperty>
    <ObjectProperty IRI="#subRoleOf"/>
</TransitiveObjectProperty>
<ReflexiveObjectProperty>
    <ObjectProperty IRI="#hasPrerequisite"/>
</ReflexiveObjectProperty>
<ReflexiveObjectProperty>
    <ObjectProperty IRI="#subConceptTypeOf"/>
</ReflexiveObjectProperty>

```

```

<ReflexiveObjectProperty>
  <ObjectProperty IRI="#subRoleOf"/>
</ReflexiveObjectProperty>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#basedOn"/>
  <Class IRI="#Personalization"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#conceptUsed"/>
  <Class IRI="#Interaction"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#consistOf"/>
  <Class IRI="#NavigationSequence"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#creates"/>
  <Class IRI="#Presentation"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#determines"/>
  <Class IRI="#Personalization"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#generates"/>
  <Class IRI="#LearningStyle"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#generates"/>
  <Class IRI="#Performance"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasCategory"/>
  <Class IRI="#LearningStyle"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasInfo"/>
  <Class IRI="#Learner"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasInteraction"/>
  <Class IRI="#Learner"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasLearningStyle"/>
  <Class IRI="#Learner"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasPerformance"/>
  <Class IRI="#Learner"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasPrerequisite"/>
  <Class IRI="#Resource"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasResource"/>
  <Class IRI="#Concept"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#hasResult"/>

```

```

    <Class IRI="#Interaction"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#hasRole"/>
    <Class IRI="#Resource"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#hasType"/>
    <Class IRI="#Interaction"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#hasType"/>
    <Class IRI="#ResourceRole"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#implements"/>
    <Class IRI="#Decision"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#isTypeOf"/>
    <Class IRI="#Resource"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#partOf"/>
    <Class IRI="#Interaction"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#subConceptTypeOf"/>
    <Class IRI="#ResourceRoleType"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#subRoleOf"/>
    <Class IRI="#ResourceRole"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#supports"/>
    <Class IRI="#Resource"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
    <ObjectProperty IRI="#basedOn"/>
    <Class IRI="#BehaviourPattern"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#basedOn"/>
    <Class IRI="#Condition"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#basedOn"/>
    <Class IRI="#CurrentGoal"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#conceptUsed"/>
    <Class IRI="#Concept"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#consistOf"/>
    <Class IRI="#Resource"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#creates"/>
    <Class IRI="#NavigationSequence"/>

```

```
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#determines"/>
  <Class IRI="#Decision"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#generates"/>
  <Class IRI="#Condition"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasCategory"/>
  <Class IRI="#LearningStyleCategory"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasInfo"/>
  <Class IRI="#PersonalInfo"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasInteraction"/>
  <Class IRI="#Interaction"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasLearningStyle"/>
  <Class IRI="#LearningStyle"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasPerformance"/>
  <Class IRI="#Performance"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasResource"/>
  <Class IRI="#Resource"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasResult"/>
  <Class IRI="#Performance"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasRole"/>
  <Class IRI="#ResourceRole"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasType"/>
  <Class IRI="#InteractionType"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasType"/>
  <Class IRI="#ResouceRoleType"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#implements"/>
  <Class IRI="#Presentation"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#isTypeOf"/>
  <Class IRI="#ResourceType"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#partOf"/>
  <Class IRI="#Session"/>
</ObjectPropertyRange>
```

```

<ObjectPropertyRange>
  <ObjectProperty IRI="#subConceptTypeOf"/>
  <Class IRI="#ResourceRoleType"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#subRoleOf"/>
  <Class IRI="#ResourceRole"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#supports"/>
  <Class IRI="#LearningStyle"/>
</ObjectPropertyRange>
<FunctionalDataProperty>
  <DataProperty IRI="#hasId"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasName"/>
</FunctionalDataProperty>
<DataPropertyDomain>
  <DataProperty IRI="#hasFileType"/>
  <Class IRI="#Resource"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasGrade"/>
  <Class IRI="#Performance"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasId"/>
  <Class IRI="#Concept"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasId"/>
  <Class IRI="#Interaction"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasId"/>
  <Class IRI="#Resource"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasLearningStyleCategory"/>
  <Class IRI="#Condition"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasLearningStyleDomain"/>
  <Class IRI="#Condition"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasName"/>
  <Class IRI="#Concept"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasName"/>
  <Class IRI="#Resource"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasRate"/>
  <Class IRI="#BehaviourPattern"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasRole"/>
  <Class IRI="#Concept"/>

```

```

</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#isRecommended"/>
  <Class IRI="#Resource"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#isVisited"/>
  <Class IRI="#Resource"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#supports"/>
  <Class IRI="#Resource"/>
</DataPropertyDomain>
<DataPropertyRange>
  <DataProperty IRI="#isRecommended"/>
  <Datatype abbreviatedIRI="xsd:boolean"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#isVisited"/>
  <Datatype abbreviatedIRI="xsd:boolean"/>
</DataPropertyRange>
<AnnotationAssertion>
  <AnnotationProperty IRI="#birthDate"/>
  <IRI>#PersonalInfo</IRI>
  <Literal datatypeIRI="&xsd;dateTimeStamp"></Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty IRI="#lastname"/>
  <IRI>#PersonalInfo</IRI>
  <Literal datatypeIRI="&xsd;string"></Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty IRI="#name"/>
  <IRI>#PersonalInfo</IRI>
  <Literal datatypeIRI="&xsd;string"></Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty IRI="#password"/>
  <IRI>#PersonalInfo</IRI>
  <Literal datatypeIRI="&xsd;string"></Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty IRI="#username"/>
  <IRI>#PersonalInfo</IRI>
  <Literal datatypeIRI="&xsd;string"></Literal>
</AnnotationAssertion>
</Ontology>

```

```

<!-- Generated by the OWL API (version 3.4.2)
http://owlapi.sourceforge.net -->

```

### **Prilog 3. *Index of Learning Styles (ILS)* upitnik**

Osnovna svrha upitnika je određivanje stilova učenja. Autori upitnika su Felder i Silverman sa univerziteta North Carolina State, USA.

Od učenika se zahteva da za svako pitanje zaokruže jedan od dva ponuđena odgovora.

Pitanja su:

1. I understand something better after I

- (a) try it out.
- (b) think it through.

2. I would rather be considered

- (a) realistic.
- (b) innovative.

3. When I think about what I did yesterday, I am most likely to get

- (a) a picture.
- (b) words.

4. I tend to

- (a) understand details of a subject but may be fuzzy about its overall structure.
- (b) understand the overall structure but may be fuzzy about details.

5. When I am learning something new, it helps me to

- (a) talk about it.
- (b) think about it.

6. If I were a teacher, I would rather teach a course

- (a) that deals with facts and real life situations.
- (b) that deals with ideas and theories.

7. I prefer to get new information in

- (a) pictures, diagrams, graphs, or maps.
- (b) written directions or verbal information.

8. Once I understand

- (a) all the parts, I understand the whole thing.
- (b) the whole thing, I see how the parts fit.

9. In a study group working on difficult material, I am more likely to

- (a) jump in and contribute ideas.
- (b) sit back and listen.

10. I find it easier

- (a) to learn facts.
- (b) to learn concepts.

11. In a book with lots of pictures and charts, I am likely to

- (a) look over the pictures and charts carefully.
- (b) focus on the written text.

12. When I solve maths problems

- (a) I usually work my way to the solutions one step at a time.
- (b) I often just see the solutions but then have to struggle to figure out the steps to get to them.

13. In classes I have taken

- (a) I have usually got to know many of the students.
- (b) I have rarely got to know many of the students.

14. In reading non-fiction, I prefer

- (a) something that teaches me new facts or tells me how to do something.
- (b) something that gives me new ideas to think about.

15. I like teachers

- (a) who put a lot of diagrams on the board.
- (b) who spend a lot of time explaining.

16. When I'm analysing a story or a novel

- (a) I think of the incidents and try to put them together to figure out the themes.
- (b) I just know what the themes are when I finish reading and then I have to go back and find the incidents that demonstrate them.

17. When I start a homework problem, I am more likely to

- (a) start working on the solution immediately.
- (b) try to fully understand the problem first.



18. I prefer the idea of

- (a) certainty.
- (b) theory.

19. I remember best

- (a) what I see.
- (b) what I hear.

20. It is more important to me that an instructor

- (a) lay out the material in clear sequential steps.
- (b) give me an overall picture and relate the material to other subjects.

21. I prefer to study

- (a) in a group.
- (b) alone.

22. I am more likely to be considered

- (a) careful about the details of my work.
- (b) creative about how to do my work.

When I get directions to a new place, I prefer

- (a) a map.
- (b) written instructions.

24. I learn

- (a) at a fairly regular pace. If I study hard, I'll "get it."
- (b) in fits and starts. I'll be totally confused and then suddenly it all "clicks."

25. I would rather first

- (a) try things out.
- (b) think about how I'm going to do it.

26. When I am reading for enjoyment, I like writers to

- (a) clearly say what they mean.
- (b) say things in creative, interesting ways.

27. When I see a diagram or sketch in class, I am most likely to remember

- (a) the picture.

(b) what the instructor said about it.

28. When considering a body of information, I am more likely to

(a) focus on details and miss the big picture.

(b) try to understand the big picture before getting into the details.

29. I more easily remember

(a) something I have done.

(b) something I have thought a lot about.

30. When I have to perform a task, I prefer to

(a) master one way of doing it.

(b) come up with new ways of doing it.

31. When someone is showing me data, I prefer

(a) charts or graphs.

(b) text summarizing the results.

32. When writing a paper, I am more likely to

(a) work on (think about or write) the beginning of the paper and progress forward.

(b) work on (think about or write) different parts of the paper and then order them.

33. When I have to work on a group project, I first want to

(a) have a "group brainstorming" where everyone contributes ideas.

(b) brainstorm individually and then come together as a group to compare ideas.

34. I consider it higher praise to call someone

(a) sensible.

(b) imaginative.

35. When I meet people at a party, I am more likely to remember

(a) what they looked like.

(b) what they said about themselves.

36. When I am learning a new subject, I prefer to

(a) stay focused on that subject, learning as much about it as I can.

(b) try to make connections between that subject and related subjects.

37. I am more likely to be considered

(a) outgoing.

(b) reserved.

38. I prefer courses that emphasise

(a) concrete material (facts, data).

(b) abstract material (concepts, theories).

39. For entertainment, I would rather

(a) watch television.

(b) read a book.

40. Some teachers start their lectures with an outline of what they will cover. Such outlines are

(a) somewhat helpful to me.

(b) very helpful to me.

41. The idea of doing homework in groups, with one grade for the entire group,

(a) appeals to me.

(b) does not appeal to me.

42. When I am doing long calculations,

(a) I tend to repeat all my steps and check my work carefully.

(b) I find checking my work tiresome and have to force myself to do it.

43. I tend to picture places I have been

(a) easily and fairly accurately.

(b) with difficulty and without much detail.

44. When solving problems in a group, I would be more likely to

(a) think of the steps in the solution process.

(b) think of possible consequences or applications of the solution in a wide range of areas.

---

## BIOGRAFIJA



- Boban Vesin je rođen 31.05.1978. u Kikindi.
- Na Prirodno-matematički fakultet u Novom Sadu, na smer Diplomirani informatičar upisao se 1997. godine, gde je i diplomirao 2002. godine sa prosečnom ocenom 8,97.
- Diplomski rad pod nazivom *Analiza funkcijskih tačaka* odbranio je 21.01.2002. godine sa ocenom 10.
- Magistarske studije upisao je 2002. godine na Prirodno-matematičkom fakultetu u Novom Sadu. Sve ispite predviđene planom i programom fakulteta položio je sa prosečnom ocenom 10. Magistarski rad pod nazivom *Dizajn i implementacija tutorskog sistema za učenje programskog jezika Java* odbranio je 15. 11. 2007 i stekao zvanje Magistra informatičkih nauka
- Na Visokoj poslovnoj školi strukovnih studija u Novom Sadu radi od 2002. godine kao predavač na predmetima *Primenjeno programiranje* i *Projektovanje sistema elektronskog poslovanja*.
- Autor je ili koautor tridesetak radova iz oblasti učenja na daljinu, sistema za generisanje preporuka, personalizacije i adaptacije u tutorskim sistemima, od kojih je pet objavljeno u časopisima sa SCI (Science citation index) liste.
- Bavi se učenjem na daljinu, programiranjem u programskom jeziku Java i bazama podataka.

Novi Sad, 2014.

UNIVERZITET U NOVOM SADU  
PRIRODNO – MATEMATIČKI FAKULTET  
KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj:

**PBR**

Identifikacioni broj:

**IBR**

Tip dokumentacije: Monografska dokumentacija

**TD**

Tip zapisa: Tekstualni štampani materijal

**TZ**

Vrsta rada: Doktorska disertacija

**VR**

Autor: Boban Vesin

**AU**

Mentor: prof. dr Mirjana Ivanović

**MN**

Naslov rada: Personalizacija procesa elektronskog učenja u tutorskom sistemu primenom tehnologija semantičkog veća

**MR**

Jezik publikacije: Srpski (latinica)

**JP**

Jezik izvoda: Srpski/engleski

**JI**

Zemlja publikovanja: Srbija

**ZP**

Uže geografsko područje: Vojvodina, Novi Sad

**UGP**

Godina: 2014.

**GO**

Izdavač: Autorski reprint

**IZ**

Mesto i adresa: Novi Sad, Trg Dositeja Obradovića 4

**MA**

Fizički opis rada: (9/140/106/13/89/0/3)

(poglavlja/strana/lit. citata/tabela/slika/grafika/priloga)

**FO**

Naučna oblast: Računarske nauke

**NO**

Naučna disciplina: Učenje na daljinu

**ND**

Ključne reči: tutorski sistemi, elektronsko učenje, Java, semantički veb, ontologije, adaptivna pravila

**PO****UDK:**

Čuva se:

**ČU**

Važna napomena: nema

**VN**

Izvod: Predmet istraživanja disertacije obuhvata realizaciju opšteg modela tutorskog sistema za elektronsko učenje iz različitih domena primenom tehnologija semantičkog veba i primena tog modela za izgradnju tutorskog sistema za učenje programskog jezika Java sa elementima personalizacije.

Cilj disertacije je implementacija i predstavljanje svih elemenata tutorskog sistema za učenje programskog jezika Java pomoću tehnologija semantičkog veba. Ovaj proces obuhvata kreiranje osnovnih gradivnih ontologija kao i pravila za izvođenje konkretnih akcija kojim se postiže personalizacija nastavnog materijala.

**IZ**

Datum prihvatanja teme od strane NN veća: 21.06.2012.

**DP**

Datum odbrane:

**DO**

Članovi komisije:

**KO**

Predsednik: dr Zoran Budimac, redovni profesor, Prirodno-matematički fakultet, Novi Sad

Mentor: dr Mirjana Ivanović, redovni profesor, Prirodno-matematički fakultet, Novi Sad

Član: dr Dragan Janković, redovni profesor, Elektronski fakultet, Niš

Član: dr Vladimir Kurbalija, docent, Prirodno-matematički fakultet, Novi Sad

UNIVERSITY OF NOVI SAD  
FACULTY OF SCIENCE KEY  
WORDS DOCUMENTATION

Accession number:

**ANO**

Identification number:

**INO**

Document type: Monograph type

**DT**

Type of record: Printed text

**TR**

Contents Code: Doctoral dissertation

**CC**

Author: Boban Vesin

**AU**

Mentor: Ph.D. Mirjana Ivanović

**MN**

Title: Personalization of learning process in Tutoring System Supported with the Semantic Web Technologies

**XI**

Language of text: Serbian (Latin alphabet)

**LT**

Language of abstract: Serbian/English

**LA**

Country of publication: Serbia

**CP**

Locality of publication: Vojvodina, Novi Sad

**LP**

Publication year: 2014.

**PY**

Publisher: Author's reprint

**PU**

Publ. Place: Novi Sad, Trg Dositeja Obradovića 4

**PP**

Physical description: (9/140/106/13/89/0/3)

**PD**

Scientific field: Computer Science

**SF**

Key words: tutoring systems, eLearning, Java, Semantic Web, ontology, adaptation rules

**UC:**

Holding data:

**Hd Note:**

Abstract: The subject of the dissertation includes the implementation of a conceptual model of tutoring system for e-learning in different domains using semantic web technologies and application of that model in a design of a tutoring system for personalised learning of Java programming language.

The goal of the dissertation is the implementation and presentation of all elements of the tutoring system for learning the Java programming language using semantic web technologies. This process includes the creation of the fundamental building blocks of ontologies and rules for carrying out the actions for adaptation of teaching materials.

**AB**

Accepted by the Scientific Board on: 21.06.2012.

Defended:

**Thesis defend board:**

President: Ph.D. Zoran Budimac, full professor, Faculty of Science, Novi Sad

Mentor: Ph.D. Mirjana Ivanović, full professor, Faculty of Science, Novi Sad

Member: Ph.D. Dragan Janković, full professor, Faculty of Electronic Engineering, Niš

Member: Ph.D. Vladimir Kurbalija, assistant professor, Faculty of Science, Novi Sad