

PRIRODOSLOVNO – MATEMATIČKI FAKULTET  
SVEUČILIŠTE U SPLITU

Marina Mihanović

# **Uporaba inteligentnih agenata u dinamičkom generiranju courseware-a**

DIPLOMSKI RAD

Split, ožujak 2009.

Studijska grupa: Matematika i informatika

Predmet: Primjena računala u nastavi

# **Uporaba inteligentnih agenata u dinamičkom generiranju courseware-a**

DIPLOMSKI RAD

Student: Marina Mihanović

Mentor: prof. dr. sc. Slavomir Stankov

Neposredni voditelj: mr. sc. Ani Grubišić

Split, ožujak 2009.

# Sadržaj

<b>1.</b>	<b>UVOD</b> .....	<b>1</b>
<b>2.</b>	<b>INTELIGENTNI TUTORSKI SUSTAVI I COURSEWARE</b> .....	<b>3</b>
<b>3.</b>	<b>INTELIGENTNI AGENTI</b> .....	<b>5</b>
3.1	OPĆENITO O AGENTIMA .....	5
3.2	VIŠEAGENTSKI SUSTAVI.....	7
3.3	INTELIGENTNI AGENTI .....	8
3.4	INTELIGENTNI AGENTI ZA PRETRAŽIVANJE .....	8
3.4.1	<i>AO* pretraživanje I-ILI grafova</i> .....	9
<b>4.</b>	<b>DINAMIČKO GENERIRANJE COURSEWARE-A</b> .....	<b>12</b>
4.1	ZAHTJEVI ZA PRILAGODLJIVI COURSEWARE.....	13
4.2	ARHITEKTURA I FUNKCIONALNOST.....	15
4.2.1	<i>Baza podataka</i> .....	16
4.2.2	<i>Model učenika</i> .....	18
4.2.3	<i>Pedagoška komponenta</i> .....	18
4.2.4	<i>Generator kolegija</i> .....	21
4.2.5	<i>Modul oblikovanja</i> .....	24
4.3	DINAMIČKO GENERIRANJE I OBLIKOVANJE KOLEGIJA .....	25
4.4	DIMENZIJE TEHNIKE OPISA DOMENE .....	26
4.5	PREDNOSTI DCG-A.....	27
4.6	IMPLEMENTACIJA DCG-A NA WWW-U.....	28
4.6.1	<i>Arhitektura DCG-a na WWW-u</i> .....	32
<b>5.</b>	<b>MODEL VIŠEAGENTSKOG SUSTAVA ZA DINAMIČKO GENERIRANJE COURSEWARE-A</b> .....	<b>34</b>
5.1	ARHITEKTURA SUSTAVA.....	34
5.2	INTELIGENTNI AGENTI U DINAMIČKOM GENERIRANJU COURSEWARE-A .....	42
5.2.1	<i>Agent sučelja</i> .....	42
5.2.2	<i>Agent modeliranja</i> .....	42
5.2.3	<i>Pedagoški agent</i> .....	42
5.2.4	<i>Agent upravljanja courseware-om</i> .....	43
5.2.5	<i>Suradnja agenata</i> .....	43
<b>6.</b>	<b>ZAKLJUČAK</b> .....	<b>45</b>
<b>7.</b>	<b>LITERATURA</b> .....	<b>46</b>

# 1. Uvod

Danas u 21. stoljeću gotovo svako kućanstvo posjeduje osobno računalo, u školstvu i akademskim zajednicama se sve više koriste računala, rad bez računala je postao gotovo nezamisliv. Kompjuteriziranost školskih, akademskih i poslovnih zajednica otvara pitanje računalne pismenosti. Zbog računalnog opismenjavanja, mnogima je potrebna prekvalifikacija za nova zanimanja ili dokvalifikacija za stara zanimanja, potrebno je stalno profesionalno usavršavanje, stoga ne čudi što je pojavom i razvojem novih tehnologija, došlo do promjena i u obrazovanju. Sastavnim dijelom sustava obrazovanja je postala informacijska i komunikacijska tehnologija (*eng. Information and Communication Technology - ICT*) kao podrška učitelju u realizaciji tradicionalne nastave ili kao zamjena takvoj nastavi s novim načinom realizacije nastavnog procesa kao i procesa učenja i poučavanja. Presjek svijeta obrazovanja i svijeta informacijske i komunikacijske tehnologije predstavlja e-učenje (*eng. e-learning*) (Stankov, 2007). Brojni autori iznose svoje viđenje e-učenja i definicije e-učenja čime se nećemo baviti u ovom radu. E-učenje je široki koncept koji obuhvaća veliki skup primjena i procesa koji upotrebljavaju sve raspoložive elektroničke medije radi isporuke obrazovanja i obuke na mnogo fleksibilniji način nego tradicionalni sustavi učenja (Stankov, 2007).

U pogledu tehnologije isporuke nastavnih sadržaja, e-učenje može biti ostvareno asinkronim ili sinkronim načinom (Stankov, 2007). *Sinkrono* učenje podrazumijeva interakciju učitelja i učenika u načelu u realnom vremenu, vremenski sinkronizirane aktivnosti koje se odvijaju po unaprijed dogovorenim scenariju na unaprijed dogovorenim mjestima. Sinkroni sustavi zadržavaju gotovo sve attribute koji obilježavaju tradicionalni predavački model poučavanja, a to su: prikaz nastavnog sadržaja s jednog mjesta, mogućnost postavljanja pitanja učitelju, ograničene mogućnosti rasprave, nastava se odvija u definiranom terminu, nastava se odvija na definiranom mjestu (Rosić, 2000). *Asinkrono* učenje je ono učenje kod kojeg se interakcija učitelja i učenika događa povremeno i u načelu s vremenskim kašnjenjem. Pri komuniciranju učitelja i učenika, njihove aktivnosti nisu vremenski sinkronizirane. Dohvat i isporuka nastavnih sadržaja moguća je na bilo kojem mjestu, u bilo kojem vremenu i napredovanje s vlastitim tempom (npr. komunikacija putem e-maila, tečajevi isporučeni putem Interneta) (Rosić, 2000). Primjer asinkronih sustava e-učenja su inteligentni tutorski sustavi koje ćemo opisati u drugom poglavlju.

Primjena *agenata* u računalnim sustavima uzrokovana je potrebama za obavljanjem zadataka u uvjetima koje obilježavaju trendovi kao što su primjerice brzi rast dostupnih informacija, raspodijeljenost kako podataka tako i obrade podataka i povećanje broja usluga koje računalni sustavi pružaju korisnicima. Kao posljedica ovog, danas se agenti koriste pri obavljanju mnogih zadataka kao što su pretraživanja sadržaja na Web-u, razvrstavanja informacija, odabira korisniku zanimljivih sadržaja, pretraživanju raspodijeljenih baza podataka i prilagodbe korisničkog sučelja raznim individualnim potrebama korisnika (Rosić, 2000).

Moguće primjene *intelligentnih agenata* koje doprinose vođenju procesa učenja i poučavanja pri preuzimanju uloge živog učitelja omogućavaju prilagodbu korisničkog sučelja individualnim potrebama korisnika, osiguravaju pomoć učenicima pri komunikaciji s drugim korisnicima sustava, pretražuju i filtriraju informacije (odgovarajući nastavni materijali, sadržaji), osiguravaju pomoć učiteljima pri poučavanju i pri oblikovanju nastavnih materijala, poučavaju učenika (odabir courseware-a za poučavanje, mijenjanje plana, tijek plana, grananje kolegija).

U drugom poglavlju se uvodi pojam inteligentnih tutorskih sustava, ključ njihovog inteligentnog ponašanja, modularna građa tih sustava te pojam i struktura courseware-a.

U trećem poglavlju se uvodi pojam agenta, dana je podjela agenata na temelju opće prihvaćene definicije agenata, kao i kratki pregled osnovne strukture agenata te svojstva (inteligentnih) agenata. Razmatraju se višeagentski sustavi kao i agenti za pretraživanje, njihovi zadaci i dvije osnovne klase algoritama za pretraživanje, s posebnim naglaskom na AO\* pretraživanje.

U četvrtom poglavlju je opisano dinamičko generiranje courseware-a - DCG, dan je pregled osnovnih razlika između računalom potpomognutog učenja i DCG-a, pregled aspekata prilagodbe individualnom učeniku, dana je arhitektura sustava s opisima komponenata te dimenzije tehnike opisa domene. Opisana je primjena DCG-a na WWW-u, kao i odgovarajuća arhitektura.

U petom poglavlju je predstavljen model višeagentskog sustava za dinamičko generiranje courseware-a, tj. primjena i doprinos inteligentnih agenata u dinamičkom generiranju courseware-a.

## 2. Inteligentni tutorski sustavi i courseware

Inteligentni tutorski sustavi (*eng. Intelligent Tutoring Systems - ITS*) su računalni sustavi namijenjeni potpori i poboljšanju procesa učenja i poučavanja u odabranom područnom znanju, uvažavajući pri tom individualnost onog koji uči i koji se poučava (Stankov, 2007). Inteligentnim tutorskim sustavima se nastoji ostvariti poučavanje „jedan na jedan“ pomoću računala, što bi značilo da inteligentni tutorski sustav preuzima ulogu živog učitelja i radom s takvim sustavom učenik stječe računalnog učitelja koji je zadužen samo za njega. Osobnost učitelja više ne igra ulogu jer nije riječ o živoj osobi, učenik je opušteniji jer uči u bilo koje vrijeme na bilo kojem mjestu bez utjecaja učitelja koji nekada znaju biti demotivirajući.

Za uvažavanje individualnosti onog koji uči i koji se poučava, ITS sustavi sadržaj i način izlaganja nastavnih sadržaja prilagođavaju individualnom učeniku i njegovim potrebama. Za uspješno izvršavanje svojih funkcionalnosti, inteligentnim tutorskim sustavima je potrebno znanje o tome što i kako poučavati te koga poučavati što ovisi o principima poučavanja te metodama i tehnikama umjetne inteligencije.

Za ostvarivanje inteligentnog ponašanja je ključno znanje pa se kaže da su inteligentni tutorski sustavi utemeljeni na znanju jer raspolažu sa (Stankov, 2007):

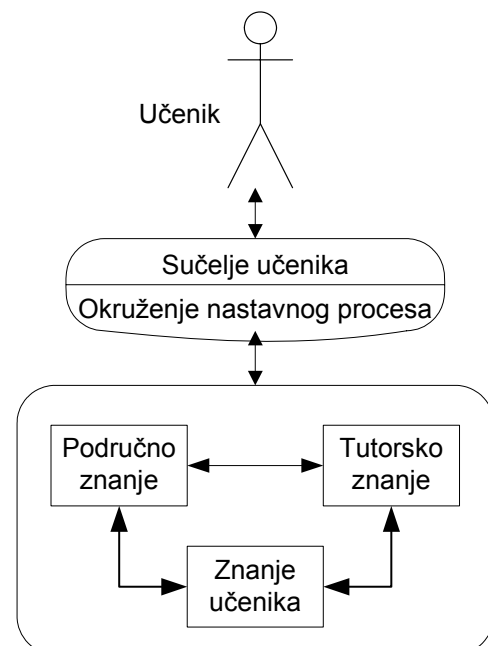
- znanjem o područnom znanju
- znanjem o principima pomoću kojih se poučava i metodama pomoću kojih se primjenjuju ti principi i
- znanjem o metodama i tehnikama za modeliranje učenika tijekom stjecanja znanja i vještina.

Inteligentni tutorski sustavi su građeni na temelju četiri međusobno povezana programska modula (Burns i Capps, prema Stankov, 2007) kako slijedi (slika 2.1):

*Ekspertni modul* (područno znanje) sadrži područno znanje koje učenik mora usvojiti tijekom procesa učenja i poučavanja. Ekspertni je modul „...kralježnica svakog inteligentnog tutorskog sustava“ (Anderson, prema Stankov, 2007).

*Modul učenika* (znanje učenika) obuhvaća sve aspekte stjecanja znanja i vještina učenika u procesu učenja i poučavanja odabranog područnog znanja, tj. omogućava modeliranje učenika, tj. oblikovanje modela učenika i dijagnozu modela učenika. Model učenika je skup podataka koji prikazuje razinu znanja i vještina učenika, a dijagnoza obrađivanjem podataka iz modela učenika vrednuje znanja učenika.

*Modul učitelja* (tutorsko znanje) preuzima ulogu živog učitelja stoga je odgovoran za vođenje procesa stjecanja znanja i vještina učenika. Odlučuje o načinu



Slika 2.1 Građa ITS-a (Stankov, 2007)

poučavanja učenika, raspolaže pedagoškim znanjem kojim inače raspolaže živi učitelj.

*Komunikacijski modul* (sučelje učenika i okruženje nastavnog procesa) predstavlja korisničko sučelje učenika i inteligentnog tutorskog sustava kojim se ostvaruje interakcija učenika, učitelja i znanja.

„**Courseware**“ je međunarodno prihvaćeni termin za nastavni sadržaj oblikovan za izvođenje na računalu (Stankov, 2007).

Nastavni sadržaji se didaktički oblikuju za određeni kolegij, no oblikuju se i za izvođenje na računalu. Courseware ima višerazinsku strukturu koja uključuje sljedeće elemente nastavnog sadržaja: *nastavne cjeline, nastavne teme, nastavne jedinice, nastavne pojmove i elemente testiranja i vrednovanja znanja učenika*. Ovakvom strukturom courseware-a je dana hijerarhija po kojoj se na najvišoj razini nalazi nastavna cjelina koja sadrži više nastavnih tema, potom nastavna tema koja sadrži više nastavnih jedinica i zatim nastavna jedinica koja sadrži više nastavnih pojmova. Elementi za testiranje i vrednovanje znanja učenika se mogu pridružiti nastavnoj cjelini, nastavnoj temi ili nastavnoj jedinici. Zadatak učitelja je oblikovanje strukture courseware-a i dekomponiranje elemenata strukture nastavnog sadržaja.

Višerazinska struktura courseware-a je u skladu sa pedagoškom tradicijom, no obzirom da je courseware predviđen za izvođenje na računalu, javljaju se novi zahtjevi zbog potrebe normiranja courseware-a. SCORM (*eng. Sharable Content Object Reference Model*) referentni model za normiranje elemenata strukture nastavnih sadržaja strukturu courseware-a temelji na objektu djeljivog nastavnog sadržaja, tzv. SCO (*eng. Sharable Content Object*). SCO je najmanji element nastavnog sadržaja koji se isporučuje učeniku za učenje, poučavanje i vrednovanje znanja i ima sljedeće atribute: ponovna upotrebljivost (*eng. reusability*), trajnost (*eng. durability*), dostupnost (*eng. accessibility*) i interoperabilnost (*eng. interoperability*). SCO može biti bilo koji od navedenih elemenata nastavnog sadržaja. Nakon strukturiranja courseware-a, potrebna su pravila za izvođenje courseware-a koja govore o tijeku i načinu isporuke courseware-a što prvenstveno ovisi o učeniku.

## 3. Inteligentni agenti

Istraživanja područja inteligentnih agenata se mogu podijeliti na razvijanje teorije o agentima, na razvijanje arhitekture agenata i na razvijanje jezika orijentiranih na agente (*eng. agent oriented languages*), kako onih u kojima su agenti napisani tako i onih koji služe za komunikaciju među agentima (Božić, prema Šuljug, 2008).

Svojstva i mogućnosti inteligentnih agenata su razlog što se smatraju novim pomakom u informacijskoj znanosti. Jedan od problema primjene inteligentnih agenata jest nedostatak standarda, tj. normiranost, no područje normi se istražuje. Unatoč preprekama koje postoje, inteligentni agenti su u širokoj primjeni, prednosti koje nude premašuju poteškoće koje stoje na putu njihove praktične primjene.

### 3.1 Općenito o agentima

Što je agent? Odgovor na to pitanje nije jedinstven. Definirati pojam agenta je podjednako teško kao i definirati pojam npr. objekt. Ne postoji jedinstvena definicija agenata, već svaki autor iznosi svoje viđenje agenta i svoju definiciju što je posljedica mladosti čitavog područja. Iako ne postoji jedinstvena definicija koja bi jednoznačno određivala agenta, agente se najopćenitije može opisati kao *entitete koji obavljaju neki posao umjesto svog vlasnika, odnosno korisnika*. Dakle, intuitivno se pri spomenu agenta podrazumijeva objekt koji samostalno djeluje u nekoj okolini, prilagođava se okolini u kojoj djeluje, ima sposobnosti percipiranja stanja u kojoj se njegova okolina nalazi, djelovanjem mijenja stanja okoline i ima sposobnost učenja (Rosić, 2000).

Opće je prihvaćena definicija agenata prema kojoj se agenti mogu podijeliti na slabu i jaku predodžbu (Wooldridge, prema Šuljug, 2008).

**Slaba predodžba agenata:** Agenti su sustavi koji imaju sljedeća obilježja:

- *Autonomnost (eng. autonomous)*. Agenti djeluju bez direktnih intervencija korisnika ili neke komponente računalnog sustava i imaju nadzor nad svojim akcijama i unutarnjim stanjem.
- *Društvenost (eng. social ability)*. Agent mora posjedovati mehanizme koji mu omogućuju koordiniranje svojih operacija s drugim agentima u svojem okruženju. Obilježje društvenosti agenta ostvareno je korištenjem komunikacijskih protokola, interakcijskih protokola i ontologija.
- *Reaktivnost (eng. reactivity)*. Reaktivni agenti ne posjeduju interni model predviđanja budućeg stanja okoline, već percipiraju svoju okolinu (koja može biti realni svijet, korisničko sučelje, kolekcija drugih agenata, Internet ili neka druga okolina) i reagiraju na promjene s akcijom predviđenom za trenutno stanje okoline i vrstu utjecaja.



- *Proaktivnost (eng. proactiveness)*. U cilju izvršavanja svog zadatka agenti ne odgovaraju samo na promjene u okolini već su sposobni preuzeti inicijativu inicirajući promjene okoline.

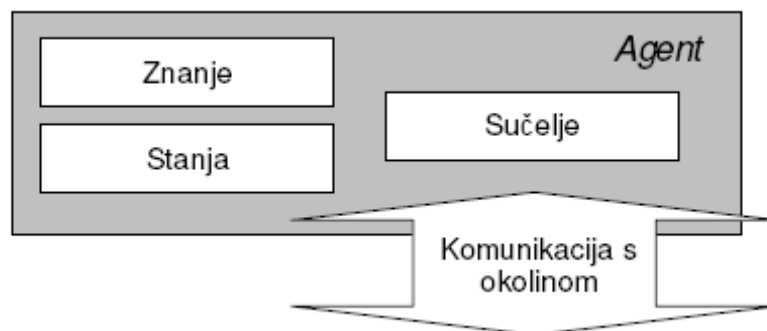
**Jaka predodžba agenata** je zastupljena u području umjetne inteligencije i ima sljedeća obilježja:

- *Pokretljivost (eng. mobile)*. Agent se može kretati s jednog mrežnog čvora na drugi. Svaki pokretni agent sastoji se od tri komponente:
  - 1) programskog koda koji sadrži logiku agenta,
  - 2) podataka, odnosno, internih atributa koji predstavljaju znanje koje agent posjeduje i
  - 3) stanja izvođenja.
- *Racionalnost (eng. rationality)*. Racionalnost definira da ako agent ima skup ciljeva, od kojih je samo jedan aktivan, on neće izvoditi akcije koje bi mogle biti u suprotnosti s njegovim trenutnim ciljevima. Racionalan agent uvijek mora izvoditi akcije koje bi u konačnici maksimizirale očekivani rezultat pri tome koristeći svoje znanje o trenutnom i budućem stanju okoline.
- *Dobronamjernost (eng. benevolence)*. Agentovi ciljevi ne smiju biti međusobno konfliktni, ako se od agenta želi da maksimizira očekivani rezultat.

Agenti ne trebaju nužno biti programski entiteti. Primjerice, robot može zadovoljiti mnoge definicije agenta. U ovom radu se pod pojmom agenta smatra isključivo programski entitet.

Osnovnu strukturu agenta je moguće prikazati kao strukturu koju čine sljedeći elementi (Rosić, 2000) (slika 3.1):

- 1) sučelje agenta
- 2) stanja agenta i
- 3) znanje agenta.

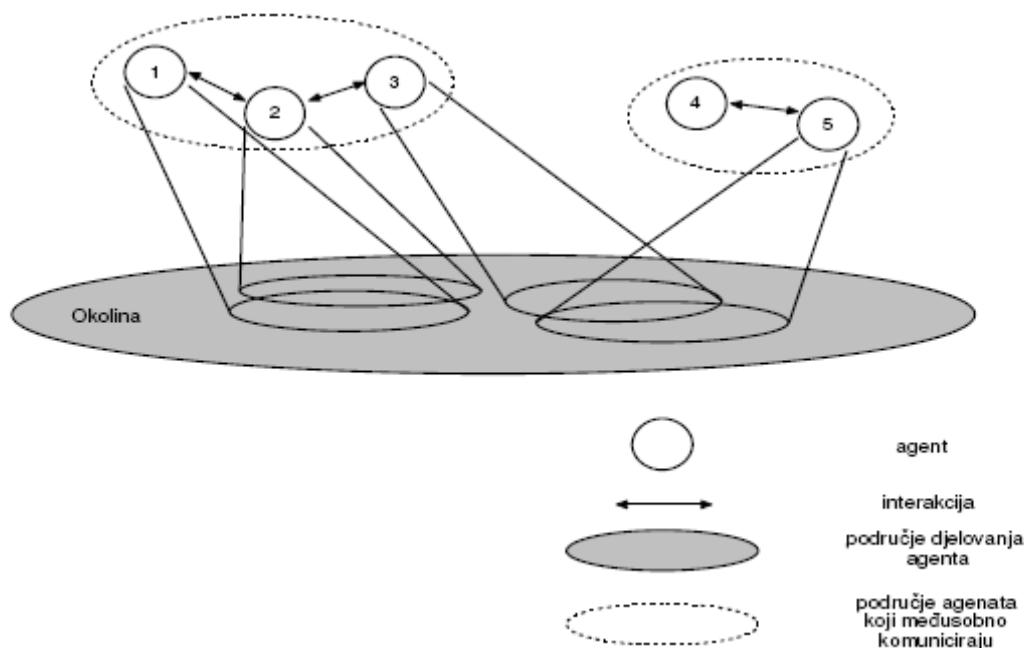


Slika 3.1 Osnovna struktura agenta (Rosić, 2000)

Interakcija agenta s njegovom okolinom zahtijeva sučelje preko kojeg agent ima mogućnost percipiranja stanja okoline i mogućnost djelovanja nad okolinom (odnosno mijenjanja stanja okoline izvođenjem agentovih akcija). Stanja agenta ovise o utjecajima okoline kao i o procesima unutar agenta, predstavljaju memoriju agenta. Znanje agenta sadrži informacije o vanjskom svijetu koje su agentu potrebne za izvršavanje njegovih zadataka kao i pravila po kojima agent djeluje (Rosić, 2000).

## 3.2 Višeagentski sustavi

Realizacijom zajedničkog rada više agenata se dobiva *višeagentski sustav*. Primjer višeagentskog sustava je prikazan na slici 3.2 (Rosić, 2000). Interakcija agenata u višeagentskom sustavu se odvija kroz njihovu komunikaciju, dijeljenje informacija kao kroz i ostale elemente zajedničke okoline. Za ostvarivanje višeagentskog sustava nije nužna interakcija svih agenata tog sustava. Agenti višeagentskog sustava dijele zajedničku okolinu, što ne znači da svaki agent ima mogućnost djelovanja na cijelu okolinu. Nema svaki agent mogućnost percipiranja stanja cijele okoline. Područja djelovanja pojedinih agenata ne moraju biti disjunktna.



Slika 3.2 Primjer višeagentskog sustava (Rosić, 2000)

Nerijetko se višeagentski sustav realizira kao otvoreni sustav unutar kojeg je dozvoljeno dinamičko dodavanje novih entiteta. Ovdje se pod entitetom podrazumijeva agent ili neki drugi element okoline. Otvoreni višeagentski sustav mora osigurati mehanizme integracije novih agenata i ostalih elemenata okoline (Rosić, 2000).

Neki autori smatraju da agenti moraju autonomno pristupiti sustavu, registrirati se i koristiti usluge sustava dok autori nekih modela predlažu da se ovaj posao dodjeli posebnim agentima ili komponentama sustava (Rosić, 2000). Agenti koji omogućavaju pristupanje sustavu, registriranje i korištenje usluga sustava moraju posjedovati informacije o stanju cijelog sustava tako da novostvorenim agentima mogu pružiti podatke o načinu interakcije s ostalim dijelovima sustava.

### 3.3 Inteligentni agenti

Često se umjesto pojma agent koristi pojam *inteligentni agent* čime se naglašava posjedovanje određene razine inteligentnog ponašanja pri obavljanju zadataka.

*Inteligencija* se opisuje različitim atributima poput sposobnosti učenja, razumijevanja, planiranja i predviđanja, rješavanja problema, sintetiziranja novih ideja i modeliranja vanjskog svijeta. Inteligenciju možemo opisati kao posjedovanje mehanizama ponašanja usmjerenih rješavanju zadanog cilja (Rosić, 2000).

*Inteligentni agenti* sadrže određeni stupanj inteligencije (nivo zaključivanja i učenja odnosno sposobnosti da prihvate korisnikove izjave o ciljevima i da obave zadatke koji se postavljaju pred njih). Inteligentni agenti su agenti koji prate naše ponašanje i na temelju njega su u stanju naučiti kako mi rješavamo probleme i sl. Kad su to naučili oni mogu te zadatke obavljati samostalno, uz povremene provjere vlasnika kod nejasnih ili novih situacija (Božić, prema Šuljug, 2008).

**Svojstva inteligentnih agenata** (Rudowsky, prema Šuljug, 2008):

- *Učenje*. Agent uči od korisnika, od drugih agenata, iz ostalih izvora.
- *Suradnja*. Agent surađuje s drugim agentima kako bi postigao svoj cilj.
- *Mobilnost*. Pokretljivost agenata po mreži, izvođenje na različitim računalima.
- *Personalizacija*. Agent poznaje svog korisnika, njegove interese i preferencije.
- *Prilagodljivost*. Agent uči iz različitih izvora te iz korisničkih akcija.

### 3.4 Inteligentni agenti za pretraživanje

U području umjetne inteligencije, česta tehnika rješavanja problema je rješavanje problema pomoću *algoritama za pretraživanje* (eng. *search algorithm*). U računalnoj znanosti, algoritam za pretraživanje u širem kontekstu je algoritam koji daje rješenje nakon vrednovanja skupa mogućih rješenja. Skup svih mogućih rješenja problema naziva se *prostor pretraživanja* (eng. *search space*) (Šuljug, 2008).

Zadatak agenta je doći iz početnog stanja (eng. *initial state*) do ciljnog stanja (eng. *goal state*) koje je rješenje problema, slijedom događaja koji uzrokuju promjene stanja. Čest je slučaj da skup stanja između početnog i ciljnog stanja nije jednoznačno određen, već postoji više različitih i različito učinkovitih putova od početnog do ciljnog stanja. Za rad takvih sustava potrebno je ugraditi algoritme za traženje spomenutih putova. Otežavajuća okolnost kod izrade spomenutih algoritama je nepostojanje formalnih postupaka koji bi na temelju početnog i ciljnog stanja sustava određivali skup stanja na putu između njih. U takvim je slučajevima potrebno pribjeći metodi *pretraživanja prostora stanja* (Bašić i ostali, prema Šuljug, 2008).

Pretraživanje prostora stanja je postupak koji slijedno prolazi kroz stanja u kojima bi se mogao nalaziti sustav te uspoređivanjem *trenutnog stanja* (eng. *current state*) s ciljnim stanjem utvrđuje da li je postupak došao do kraja. Skup stanja organiziran je u strukturu grafa pa se pretraživanje svodi na pronalaženje puta kroz graf. Čvorovi grafa pritom predstavljaju stanja sustava, a dva su čvora povezana granom u grafu ukoliko se iz jednog stanja može prijeći u drugo. Svaki čvor ima vezu na čvor prethodnik pa se može pronaći put u grafu kad se

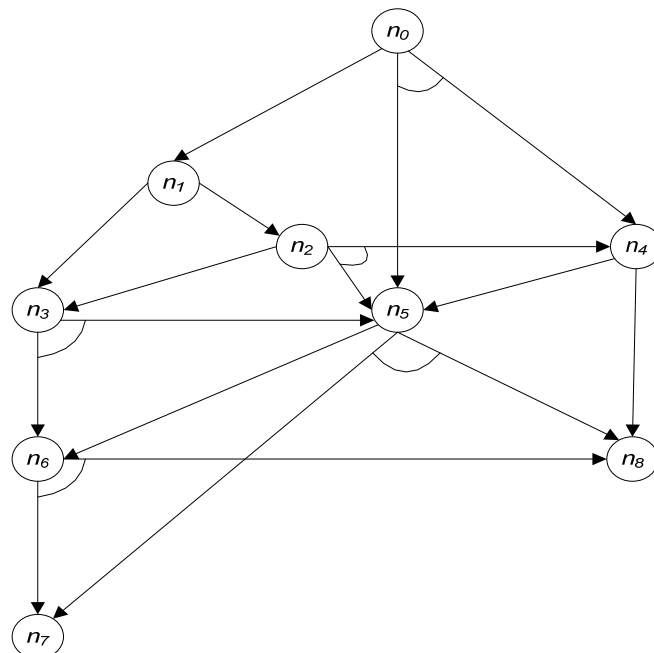
pronade konačno stanje. Put se gradi tako se krene od ciljnog čvora, te se slijedeći veze na prethodnike, dođe do početnog čvora grafa (Bašić i ostali, prema Šuljug, 2008).

Razlikujemo dvije osnovne klase algoritama za pretraživanje: slijepo ili neinformirano pretraživanje (*eng. blind search or uninformed search*) te heurističko ili usmjereno (informirano) pretraživanje (*eng. heuristic or informed search*) (Youngblood, prema Šuljug, 2008). *Slijepo pretraživanje* je pretraživanje koje nema nikakvih informacija o broju koraka ili vrijednosti putanje (vrijednost udaljenosti čvora od početnog čvora) od početnog do krajnjeg stanja, tj. cilja (npr. pretraživanje po dubini, po širini). *Heurističko pretraživanje* je pretraživanje koje ima dodatne informacije o cilju, vrijednost putanje ili broj koraka. Te informacije čine ova pretraživanja boljim od slijepih te im omogućuju gotovo racionalno ponašanje. U ova pretraživanja spada AO\* (*eng. AND-OR star search*) algoritam za pretraživanje čija će se primjena opisati u četvrtom poglavlju. U literaturi se javljaju razne varijante AO\* algoritma koje ovise o dodatnoj informaciji, tj. o odabranoj heurističkoj funkciji.

### 3.4.1 AO\* pretraživanje I-ILI grafova

AO\* algoritam za pretraživanje je poseban slučaj algoritma pretraživanja po širini za pretraživanje I-ILI grafova (Nilsson, 1980). I-ILI grafovi (*eng. AND-OR graphs*) su korisni za određene probleme čije rješavanje uključuje dekomponiranje problema na više manjih pod-problema tako da se svi ili samo jedan od tih pod-problema moraju riješiti kako bi došli do rješenja originalnog problema.

Svaki problem je prikazan čvorom (*eng. node*) u usmjerenom grafu gdje konektori<sup>1</sup> (*eng. connectors*) odražavaju odnos problema i pod-problema. Pojavljuju se dvije vrste čvorova vezanih uz dvije vrste dekomponiranja, I čvorovi (*eng. AND nodes*) i ILI čvorovi (*eng. OR nodes*), a dobivena struktura je I-ILI graf. Na slici 3.3 je prikazan primjer I-ILI grafa.



Slika 3.3 Primjer I-ILI grafa (Nilsson, 1980)

<sup>1</sup> Umjesto lukova (*eng. arcs*) koji povezuju parove čvorova, koriste se tzv. hiper-lukovi (*eng. hiperarcs*) koji povezuju roditeljski čvor sa skupom čvorova – sljedbenika (*eng. successor*). Ti hiper-lukovi se zovu konektori (*eng. connectors*).

Labela čvora ( $I$  ili  $ILI$ ) ovisi o vrsti veze čvora i njegovog roditeljskog čvora. Obzirom da jedan čvor može imati više roditeljskih čvorova, to jedan te isti čvor može biti i  $I$  čvor i  $ILI$  čvor u ovisnosti o odabranom roditeljskom čvoru.

Određeni pod-problemi opisani u grafu kao *terminalni čvorovi* (*eng. terminal nodes*) se mogu izravno riješiti i rješenje originalnog problema (njemu odgovara korijenski čvor) se može pronaći primjenom tehnika obilaženja grafa. Graf rješenja (*eng. solution graph*) se sastoji od pod-grafova koji povezuju korijenski čvor sa jednim ili više terminalnih čvorova. Terminalni čvorovi su listovi (čvorovi bez sljedbenika). Još jedna vrsta čvorova su *neterminalni listovi* (*eng. nonterminal leaves*) koji predstavljaju osnovne pod-probleme koji nemaju rješenja. Takvi čvorovi se očito ne mogu pojaviti u rješenju pod-grafa.

Svaki  $k$ -konektor je usmjeren iz roditeljskog čvora do skupa  $k$  sljedbenika. Na slici se  $k$ -konektor,  $k \geq 2$  označava zakrivljenom linijom koja spaja lukove od roditeljskog čvora do elemenata skupa sljedbenika.

Konektorima I-ILI grafa se pridjeljuje cijena (*eng. cost*) koja se može koristiti za računanje cijene grafa rješenja. Cijena grafa rješenja od bilo kojeg čvora  $n$  do skupa čvorova  $N$  se označava sa  $k(n, N)$  i može se zadati rekurzivno kako slijedi (Nilsson, 1980):

- Ako je  $n$  element skupa  $N$ , onda je  $k(n, N) = 0$ .
- Inače  $n$  ima izlazni konektor do sljedbenika, elemenata skupa  $N = \{n_1, \dots, n_i\}$  grafa rješenja. Neka je cijena ovog konektora  $c_n$ .

Tada je  $k(n, N) = c_n + k(n_1, N) + \dots + k(n_i, N)$ .

Procedura AO\* koristi heurističku funkciju  $h$  koja daje procjenu cijene optimalnog grafa rješenja od čvora  $n$  do skupa terminalnih čvorova (Nilsson, 1980):

1. Kreiraj graf pretraživanja  $G$ , koji se sastoji samo od početnog čvora  $s$ . Pridruži čvoru  $s$  cijenu  $q(s) = h(s)$ . Ako je  $s$  terminalni čvor, označi ga sa RIJEŠEN.
2. Dok  $s$  nije označen sa RIJEŠEN ponavljaj:
  - 2.1. izračunaj djelomično rješenje grafa  $G'$  u  $G$  praćenjem označenih konektora u  $G$ -u od  $s$
  - 2.2. odaberi bilo koji neterminalni list  $n$  od  $G'$
  - 2.3. proširi čvor  $n$  stvaranjem svih njegovih sljedbenika i postavi ih u  $G$  kao sljedbenike od  $n$ .
    - 2.3.1. Za svaki sljedbenik  $n_j$  koji se još nije pojavio u  $G$ , pridruži cijenu
 
$$q(n_j) = h(n_j)$$
    - 2.3.2. Sljedbenike koji su terminalni čvorovi označi sa RIJEŠEN.
  - 2.4. Kreiraj jednočlani skup čvorova  $S$  koji sadrži samo čvor  $n$ .
  - 2.5. Dok  $S$  nije prazan skup, ponavljaj:
    - 2.5.1. početak
    - 2.5.2. ukloni iz skupa  $S$  čvor  $m$  takav da nema potomaka u  $G$ , a javlja se u  $S$
    - 2.5.3. preispitaj cijenu  $q(m)$  kako slijedi:
      - 2.5.3.1. Za svaki konektor usmjeren iz  $m$  do skupa čvorova  $\{n_{li}, \dots, n_{ki}\}$  izračunaj:
 
$$q_i(m) = c_i + q(n_{li}) + \dots + q(n_{ki})$$

- 2.5.3.2. Postavi  $q(m)$  na minimum od svih izlaznih konektora iz  $q_i(m)$  i obilježi konektor kojim je taj minimum ostvaren. Ako su svi sljedbenici kroz ovaj konektor označeni kao RIJEŠEN-i, označi čvor  $m$  sa RIJEŠEN.
- 2.5.4. Ako je  $m$  označen sa RIJEŠEN ili ako je preispitana cijena različita od prethodne, u skup  $S$  dodaj sve one roditelje od  $m$  takve da je  $m$  jedan od njihovih sljedbenika kroz obilježeni konektor.

## 4. Dinamičko generiranje courseware-a

Zadnja dva desetljeća dvadesetog stoljeća, polje računalom potpomognutog učenja (*eng. Computer Aided Learning - CAL*) se uglavnom razvija u smjeru uključivanja novog prikaza medija u stari koncept courseware-a. U isto vrijeme, polje inteligentnih tutorskih sustava je doživjelo intenzivan rast, uspješnost i neka razočaranja vezana uz namjenu za praktične primjene. Međutim, razvoj ITS sustava nije utjecao na tradicionalan koncept CAL-a. Tek se krajem dvadesetog stoljeća uvidjela potreba za spajanjem ova dva polja pa je bilo pokušaja intelektualizacije i obrade CAL-a. Prvi i najvažniji korak je dinamičko generiranje prilagodljivog courseware-a. Učenikovo razumijevanje ključno za svaki korak courseware-a i ponekad je potrebno puno više fleksibilnosti nego što je to moguće ostvariti u sklopu tradicionalnog CAL courseware-a. Razvijen je alat za oblikovanje prilagodljivih kolegija (*eng. course*) nazvan *dinamičko generiranje courseware-a* (*eng. Dynamic Courseware Generation - DCG*) koji generira individualne nastavne sadržaje u skladu sa učiteljevim ciljevima i predznanjem učenika te dinamički prilagođava kolegij na osnovu učenikovog uspjeha u procesu stjecanja znanja i vještina.

Novi pristup pruža alternativu standardnom CAL oblikovanju i temelji se na drukčijem poimanju kolegija koji se za vrijeme procesa učenja i poučavanja dinamički generira kako bi što bolje pristajao napretku i potrebama individualnog učenika. Učitelj aktivno sudjeluje u stvaranju courseware-a bez potrebe poznavanja autorskog alata. U aktualnom procesu oblikovanja courseware-a, autor mora eksplicitno predstaviti strukturu domene (koncepte i veze) za poučavanje odnosno učenje. Sustav to znanje koristi kako bi generirao različite planove kolegija za različite ciljeve te kako bi ih izvršio izdvajanjem prezentacijskih materijala ovisno o učiteljevoj nastavi i učenikovim potrebama.

DCG se temelji na arhitekturi ITS ljuske čija je glavna zamisao globalno planiranje nastavnih sadržaja. Sustav dinamički generira nastavne kolegije na osnovu zasebnih eksplicitnih prikaza strukture domene i biblioteke nastavnih materijala. Plan kolegija se zasebno kreira za danog učenika i dane ciljeve poučavanja, potkrijepljen je nastavnim materijalima i može se dinamički mijenjati u skladu sa potrebama učenika. Glavna prednost ovog pristupa je mogućnost automatske izgradnje fleksibilnih CAL kolegija, što nije moguće u sklopu tradicionalnog CAL courseware-a. Zbog odvojenosti strukture znanja od njegovog prikaza, podržano je oblikovanje ponovnom upotrebom postojećih materijala, kao i proširenje i mijenjanje postojećih materijala.

(Van Marcke, prema Vassileva, 1998) je predložio pristup GTE (*eng. Generic Teaching Expertise*) za premošćivanje razlika između CAL-a i ITS sustava koji ne koristi standardne tehnike umjetne inteligencije za intelektualizaciju CAL-a, već zauzima perspektivu utemeljenu na zadatku (*eng. task-based perspective*) definiranjem hijerarhije obrazovnih zadataka i modeliranjem nastave izvođenjem redoslijeda zadataka. DCG se kombinira sa GTE za pružanje dinamičkog planiranja i sadržaja i isporuke u nastavi. GTE osigurava metodu za određivanje redoslijeda nastavnih materijala, tj. usredotočen je na problem *kako poučavati, a ne što poučavati* (Vassileva, 1998).

Organiziranjem strukture domene tako da se razlikuju različiti aspekti, moguće je koristiti različite diskurzivne strategije za generiranje planova sa suvislim sadržajima. Djelomično generirano pedagoško znanje, koje je eksplicitno prikazano pomoću hijerarhije obrazovnih zadataka i pravila poučavanja osigurava pedagošku konzistentnost automatski generiranih kolegija.

Iako posjeduje određene inteligentne značajke, ovaj sustav nije ITS sustav obzirom da nema dijagnoze učenikovih odgovora, dinamičku domenu i poučavanje stručnosti. Vjeruje se da je razuman kompromis između ITS sustava i CAL-a koji uglavnom rezultira u naprednom oblikovanju koncepata obzirom da se kolegij automatski generira. Oblikovanje kolegija je podijeljeno između autora (projektant baze podataka određene domene) i učitelja čime se učitelje aktivno uključuje u kreiranje kolegija bez previše truda i posebnog znanja o oblikovanju i autorskom alatu i pomaže u sveukupnom odobravanju sustava.

## 4.1 Zahtjevi za prilagodljivi courseware

Kolegij generiran DCG-om učenicima izgleda kao CAL kolegij. Međutim, ovaj kolegij se generira posebno za svakog učenika kako bi se ostvarili određeni ciljevi poučavanja nekog sadržaja (tema ili koncept koji se mora naučiti) i uzima u obzir postojeće znanje i preferencije određenog učenika. Kolegij je dinamičan, tj. mijenja se za vrijeme izvršavanja i prilagođava se napretku učenika, njegovom stilu učenja i preferencijama. Glavne razlike između CAL i DCG pristupa su sažete u tablici 4.1.

Tablica 4.1 Razlike CAL-a i DCG-a (modificirano prema (Vassileva, 1997))

Razlike u:	CAL	DCG
<b>Definiranje, odabir kolegija</b>	Predefinirani kolegij sa fiksnim ciljem poučavanja	Kolegij usmjeren na cilj se generira pri pokretanju
<b>Početna točka</b>	Fiksna početna točka ili izbor od nekoliko početnih točaka	Odabire se u odnosu na učenikovo postojeće znanje
<b>Prikaz materijala</b>	Fiksni redoslijed	Dinamički se odabire u odnosu na strategiju poučavanja i učenikove preferencije
<b>Praćenje kolegija</b>	Predefinirani redoslijed u skladu sa napretkom hipotetskog prosječnog učenika	Automatsko ponovno planiranje pri izvršavanju kako bi odgovarali individualnim razlikama u znanju i strategijama poučavanja
<b>Uloga učitelja</b>	Isključena (osim ako ne preuzme ulogu autora)	Dodjeljuje cilj poučavanja i može upravljati strategijom poučavanja uređivanjem skupa pravila poučavanja

U dinamičkom generiranju courseware-a je kao i u tradicionalnom CAL courseware-u unaprijed određen tijek interakcije. Prilagodba individualnom učeniku se događa u sljedećim aspektima (Vassileva, 1994):

- *U odabiru kojeg se kolegija pridržavati.* Kolegij se generira kako bi bio prikladan za poučavanje ciljeva određenog učenika.
- *U odabiru polazišnog kolegija za određenog učenika na osnovu njegovog postojećeg znanja.*
- *U odabiru prezentacijskih materijala adekvatnih za određenog učenika.*



- *U tijeku kolegija.* U CAL sustavima, kolegij se određuje a priori, pretpostavljajući da je razina znanja znanje hipotetskog „prosječnog“ učenika. Postojeća razina znanja određenog učenika se ne uzima u obzir, kao ni razvoj tijekom courseware-a. U dinamičkom generiranju courseware-a, kolegij se u početku generira u skladu sa ciljevima određenog poučavanja, ali sustav može donositi zaključke o inicijalnom statusu i razvoju učenikove razine znanja te mijenjati kolegij u skladu sa takvim razvojem.
- *U načinu grananja.* Dok je u CAL courseware-u grananje kolegija direktan odgovor na učenikovu pogrešku, sustav bi vrednovanjem učenikove razine znanja trebao prosuditi javlja li se i kada, potreba za grananjem ili mijenjanjem kolegija. Opće mišljenje je da sustav mora biti reaktivan, no smatra se da je to ipak pedagoška odluka kompetencije učitelja koji će se koristiti sustavom u svom razredu, ili kompetencije samih učenika koja će se dinamički mijenjati tijekom kolegija. Odluka se ne bi trebala donijeti jednom zauvijek i to u fazi stvaranja kolegija.

Razmatra se drugi aspekt prilagodljivosti: mogućnost prilagodbe courseware-a učiteljevim željama, ciljevima i prednostima. Tradicionalni koncept CAL courseware-a isključuje učitelja, osim u slučaju preuzimanja uloge autora i pisanja vlastitog courseware-a, što je glavna prepreka široj primjeni CAL-a u tradicionalnim postavkama školstva. Učitelju bi se trebalo omogućiti generiranje courseware-a u skladu s njegovim željama bez potrebe za oblikovanjem.

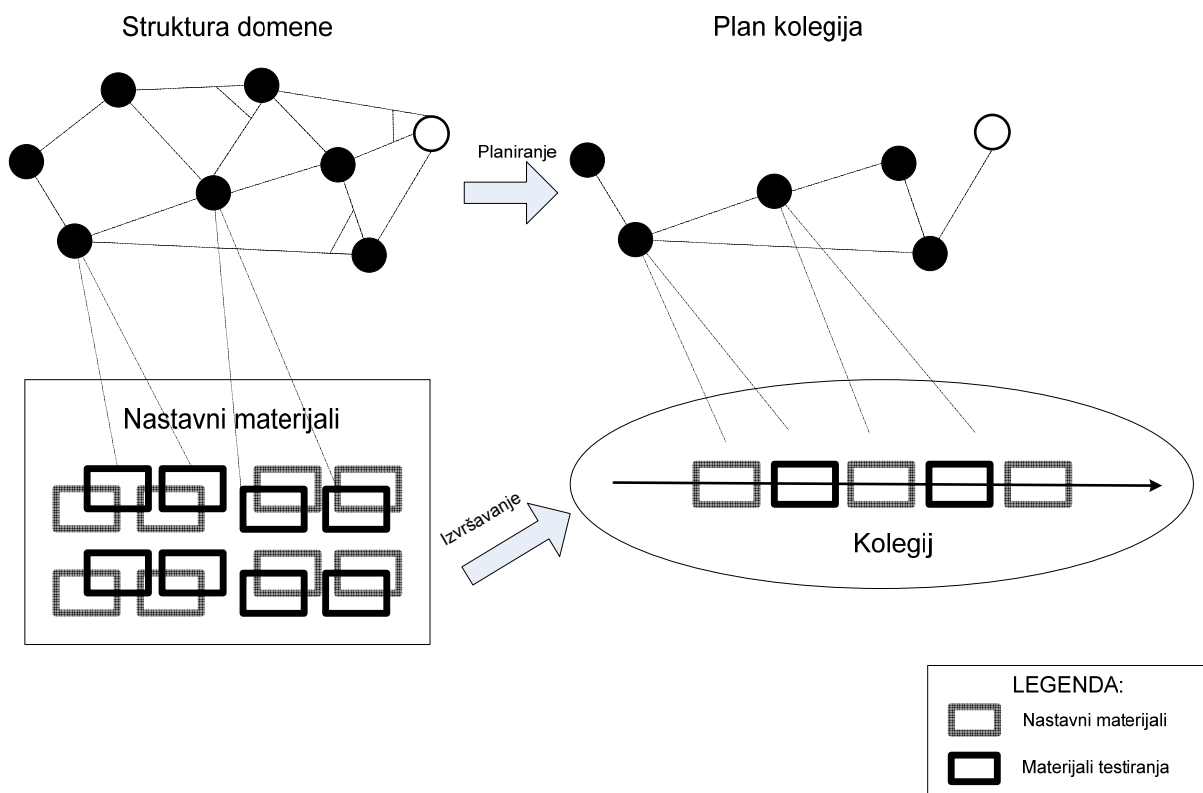
Osim prethodno navedenih razlika između tradicionalnog CAL courseware-a i novog prilagodljivog courseware-a, javljaju se još neke razlike zbog specifičnosti poučavanja naprednih i složenih tehničkih sustava (Vassileva, 1994).

*Poučavanje istih sadržaja s različitim ciljevima poučavanja pomoću različitih gledišta.* Aktualni tehnički sustavi su toliko složeni da ih je nemoguće iscrpno opisati bez uzimanja u obzir različitih aspekata i razina apstrakcije. Primjerice, uređaj se može opisati sa konstrukcijskog, funkcionalnog i geometrijskog gledišta, ili različitim kombinacijama ovih temeljnih gledišta. U sklopu tradicionalnog CAL courseware-a, to se može ostvariti samo kreiranjem različitih kolegija za svako moguće gledište i/ili svaku kombinaciju temeljnih gledišta.

*Održavanje centraliziranog prikaza znanja vezanog uz uređaje radi lakšeg ažuriranja bez utjecaja na ostatak znanja sustava.* Takvo održavanje je neophodno za napredne tehničke sustave koji se često mijenjaju. U sklopu tradicionalnog CAL courseware-a, kolegij se mora intenzivno analizirati i mijenjati, uzimajući u obzir njihovu potencijalnu djelotvornost i odgovarajuće promjene kroz kolegij. Obzirom da nema alata koji pomažu sačuvati konzistentnost kolegija pri uvođenju promjena, to je teško ostvariti pa nam je potreban centraliziran prikaz znanja vezanog uz uređaje koji se može lako ažurirati. S druge strane, potreban nam je prikaz nastavnih materijala koji će to znanje izložiti učeniku. Baza nastavnih materijala sadrži polivalentne „atome“ koji se mogu kombinirati kako bi formirali „molekule“ koje se tada mogu kombinirati za formiranje kolegija nastave za postizanje danih ciljeva poučavanja.

## 4.2 Arhitektura i funkcionalnost

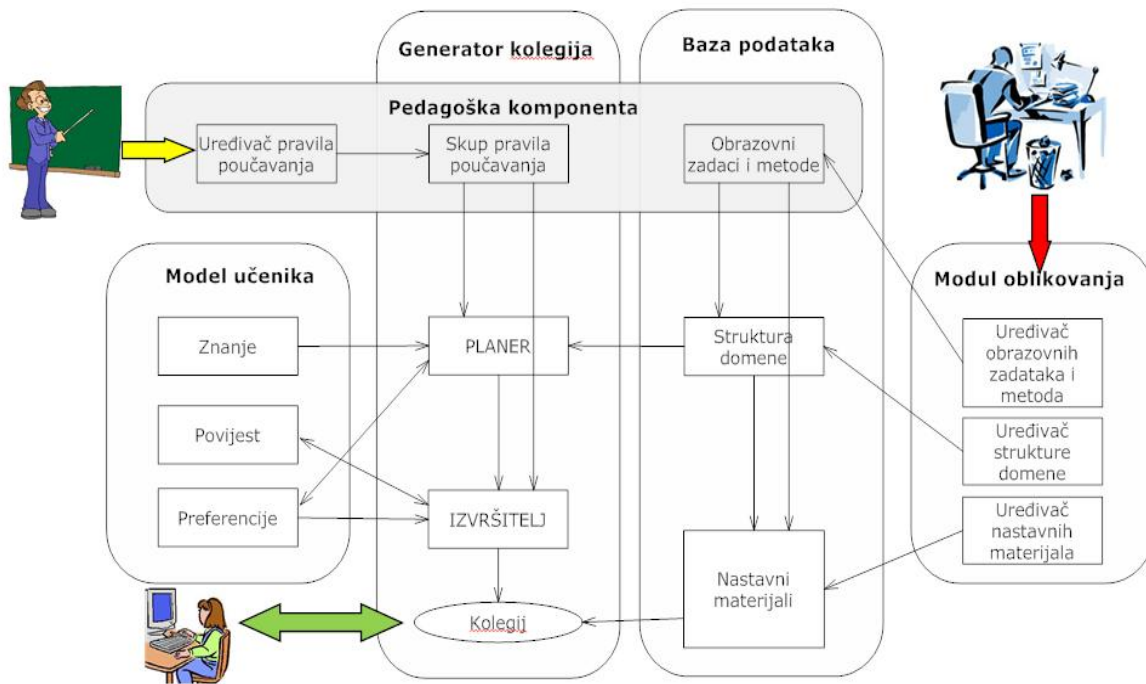
DCG je realiziran kombinacijom arhitekture ITS ljuske koja dinamički generira plan kolegija u skladu sa danim ciljem i GTE<sup>2</sup> arhitekture (Van Marcke, 1991) koja na osnovu skupa hijerarhije obrazovnih zadataka i metoda odlučuje o načinu iznošenja optimalnog plana za učenika u skladu sa pravilima poučavanja. Glavna značajka DCG arhitekture je *definiranje različitih razina prikaza znanja o domeni što omogućava odvajanje postojeće strukture domene od strukture ciljeva poučavanja koja ovisi o pojedinoj sesiji poučavanja i o učeniku*. Ovakvo odvajanje omogućava učitelju da eksplicitno definira način na koji želi poučavati ciljeve vezane uz odgovarajući koncept, koje vrste veza odgovarajućih koncepata slijediti. Ideja DCG-a je korištenje klasičnih mehanizama umjetne inteligencije za planiranje prikaza znanja domene u I-ILI grafu za automatsko generiranje sadržaja plana kolegija (slika 4.1).



Slika 4.1 Glavna ideja DCG-a (modificirano prema (Vassileva i Deters, 1998))

DCG se sastoji od sljedećih komponenti: *baza podataka* (nastavni materijali i struktura domene), *model učenika*, *generator kolegija* (planer i izvršitelj) i *modul oblikovanja* (uređivač nastavnih materijala, uređivač strukture domene) koje će se u ovom poglavlju detaljnije opisati. (Vassileva, 1994). U dodatku GTE-a DCG arhitekturi, nova arhitektura sadrži i *pedagošku komponentu*, a modul oblikovanja u dodatku sadrži i uređivač obrazovnih zadataka i metoda. Na slici 4.2 je prikazana arhitektura sustava (DCG + GTE).

<sup>2</sup>GTE (*eng. Generic Teaching Expertise*) se temelji se na prikazu poučavanja stručnosti (*eng. teaching expertise*) u terminima obrazovnih zadataka (*eng. instructional tasks*), metoda (*eng. instructional methods*) i objekata (*eng. instructional objects*) koji omogućavaju navigaciju strukturom sadržaja, razradu individualnog kolegija, upravljanje vježbama i vrednovanje napretka učenika (Mispelkamp i Sarti, 1995).



Slika 4.2 Arhitektura sustava (DCG+GTE) (modificirano prema (Vassileva, 1997))

DCG ne rješava problem *kako* učeniku prikazati odabrani sadržaj (trenutni koncept ili vezu). Za vrijeme izvršavanja, odabire jedan ili više nastavnih materijala nastojeći se koristiti onim nastavnim materijalima čija se pedagoška vrsta dotad pokazala uspješnima. Obzirom da su takve vrijednosti nastavnih materijala diskretne, nema glatkog prijelaza između prikaza, ne postoji mogućnost prikaza materijala u skladu sa strategijom poučavanja. Ovaj problem se rješava povezivanjem DCG-a i GTE-a, što sustavu daje znanje o tome *kako* poučavati dani sadržaj (prikazan obrazovnim zadacima i metodama) i dopušta planiranje prikaza već odabranog sadržaja (trenutni ciljni koncept iz plana sadržaja) na pedagoški značajan način.

DCG odlučuje o tome koje će se koncepte poučavati, tj. dinamički kreira plan sadržaja kolegija, a GTE pruža prikaz obrazovnih zadataka i metoda što sustavu omogućuje dinamičko planiranje prikaza sadržaja vezanog uz trenutni koncept, na optimalan način za učenika, tj. vrstu i redosljed nastavnih materijala (Vassileva, 1998).

### 4.2.1 Baza podataka

Predmet znanja je sadržan u komponenti baza podataka (*eng. data base*) koja se sastoji od nastavnih materijala i strukture domene (Vassileva, 1997).

#### NASTAVNI MATERIJALI

Nastavni materijali (*eng. teaching materials*) sadrže jedinice za prikaz i testiranje znanja koje vrše komunikaciju sa učenikom, tj. to je ono što učenici vide na ekranu. Usredotočeni su na dani element znanja (koncept) ili na objašnjenje o konkretnoj vezi između dva elementa znanja. Razumljivi su sami po sebi i u kombinaciji stvaraju kolegij.

Nastavni materijali se mogu klasificirati obzirom na njihovu *pedagošku funkciju*, primjerice uvod u koncept, problem motivacije, objašnjenje, pomoć, vježba ili test. U ovom smislu, nastavni materijali su ekvivalentni osnovnih funkcija obrazovanja. Nastavni materijali koji

iznose komunikaciju sa učenikom (vježbe, testovi) su predstavljeni skupom manjih jedinica koje pružaju prethodno pohranjene točne odgovore na pitanja vježbi odnosno testova, ili upute, objašnjenja te eventualno korake rješavanja problema. Nastavni materijali pedagoške vrste „test“ u dodatku imaju dvije pridružene vjerojatnosti koje označavaju do koje mjere učenikov točan (netočan) odgovor znači da učenik zna (ne zna) koncepte obuhvaćene testom. Nastavni materijali se mogu klasificirati i obzirom na *korišteni medij*, tj. tekstualni, grafički, animirani ili video-zapis nastavnih materijala, tj. mogu biti bilo koji način vizualizacije znanja kojeg se mora poučavati.

Nastavni materijali ne sadrže one činjenice za koje se očekuje da će se mijenjati. Ova informacija je pohranjena u strukturi domene gdje se može lako ažurirati. Nastavni materijali sadrže samo one parametre koji će tijekom izvršavanja kolegija poprimiti stvarne vrijednosti.

## STRUKTURA DOMENE

Struktura domene (*eng. domain structure*) sadrži strukturu predmeta znanja koncepta (odnosno veze) kojeg će se poučavati. Prikazana je kao I-ILI graf sa *čvorovima* (*eng. nodes*) koji odgovaraju elementima znanja (konceptima, pravilima i sl.) i *vezama* (*eng. links*) koje odgovaraju mogućim relacijama među čvorovima.

Ako su dva čvora A i B povezana sa čvorom C sa *I-vezom*, to znači da se oba čvora A i B moraju razviti slijedeći ovu vezu od čvora C, inače će se smatrati alternativama, tj. postoji izbor koji će se čvor razviti, ili A ili B (*ILI-veza*).

Veze mogu imati razne semantike. Primjerice, ako su čvorovi A i B povezani sa čvorom C *I-vezom* vrste *agregacija*, to znači da C sadrži pod-komponente A i B. Ako su povezani *ILI-vezom* vrste *generalizacija*, to znači da je C općeniti koncept sa mogućom instancom A ili B. Postoje različite semantičke relacije između dva koncepta, primjerice: agregacija, generalizacija, analogija, implikacija.

Najjednostavniji način definiranja strukture domene je korištenje samo jedne semantičke veze, primjerice povezati koncepte domene semantičkom vezom – *preduvjet* koja kodira pedagoški slijed. Na ovaj način možemo dobiti strukturu poput kurikuluma koja služi vođenju slijeda sadržaja (Peachey i McCalla, prema Vassileva, 1997). Ovaj pristup se u literaturi javlja pod različitim imenima: model sadržaja (Van Marcke, prema Vassileva, 1997), pedagoška struktura domene (Vassileva, 1997).

I-ILI grafovi su odabrani kao prezentacijski formalizam, obzirom da imaju jaku izražajnu moć, ekvivalent su sustava produkcijskih pravila dekomponiranja (Nilsson, 1980). I-ILI grafovi se mogu vizualizirati na ekranu, što ima psiholoških prednosti za oblikovanje i poučavanje. Moguće je organizirati koncepte domene u skup manjih, po mogućnosti međusobno vezanih I-ILI grafova, koji prikazuju relativno nezavisna pod-područja domene, ili različita gledišta. Takve pod-grafove nazivamo *aspektima*.

Svakom čvoru odnosno vezi strukture domene je pridružen skup različitih nastavnih materijala obzirom na pedagošku vrstu ili medije, koji predstavljaju taj koncept odnosno vezu. Struktura domene se koristi za kreiranje plana sadržaja kolegija (pod-graf strukture domene) koji će sadržavati dane ciljeve poučavanja (koncept). Plan se naziva *plan sadržaja* (*eng. content plan*), a proces se naziva *planiranje sadržaja* (*eng. content planning*). Tijekom izvršavanja kolegija, odabir nastavnih materijala za poučavanje koncepta plana se vrši različitim zadacima poučavanja.

## 4.2.2 Model učenika

Model učenika (*eng. student model*) se sastoji iz tri dijela (Vassileva, 1997):

- model znanja učenika (*eng. student knowledge*)
- povijest (*eng. history*)
- model učenikovih osobnih poteza i preferencija (*eng. personal traits and preferences*)

*Model znanja učenika* je pokriven strukturom domene u bazi podataka domene, sadrži vjerojatnostna vrednovanja uvjerenja da je učenik usvojio dani koncept. Ažuriranje vjerojatnosti znanja se izvodi dinamički kao posljedica učenikovih točnih odnosno netočnih odgovora na pitanja testa. (Ovo nije originalna tehnika modeliranja učenika.)

*Povijest* sadrži listu obrazovnih zadataka i metoda te nastavnih materijala korištenih za neki koncept tijekom praćenja plana. Pohranjuje i statistike o uspjehu različitih metoda dekomponiranja obrazovnog zadatka i statistike o uspjehu različitih vrsta medija (tekst, zvuk, grafički zapis, animacija, video zapis).

*Model učenikovih osobnih poteza i preferencija* sadrži dvije liste varijabli s vrijednostima. Prva lista označuje psihološke značajke poput motivacije, koncentracije, pozornosti i slično, a druga lista sadrži vrste medija koje učenik preferira. Vrijednosti mogu poprimiti tri diskretne vrijednosti slabo, umjereno i visoko i dodijeljene su od strane učenika na početku sesije poučavanja.

## 4.2.3 Pedagoška komponenta

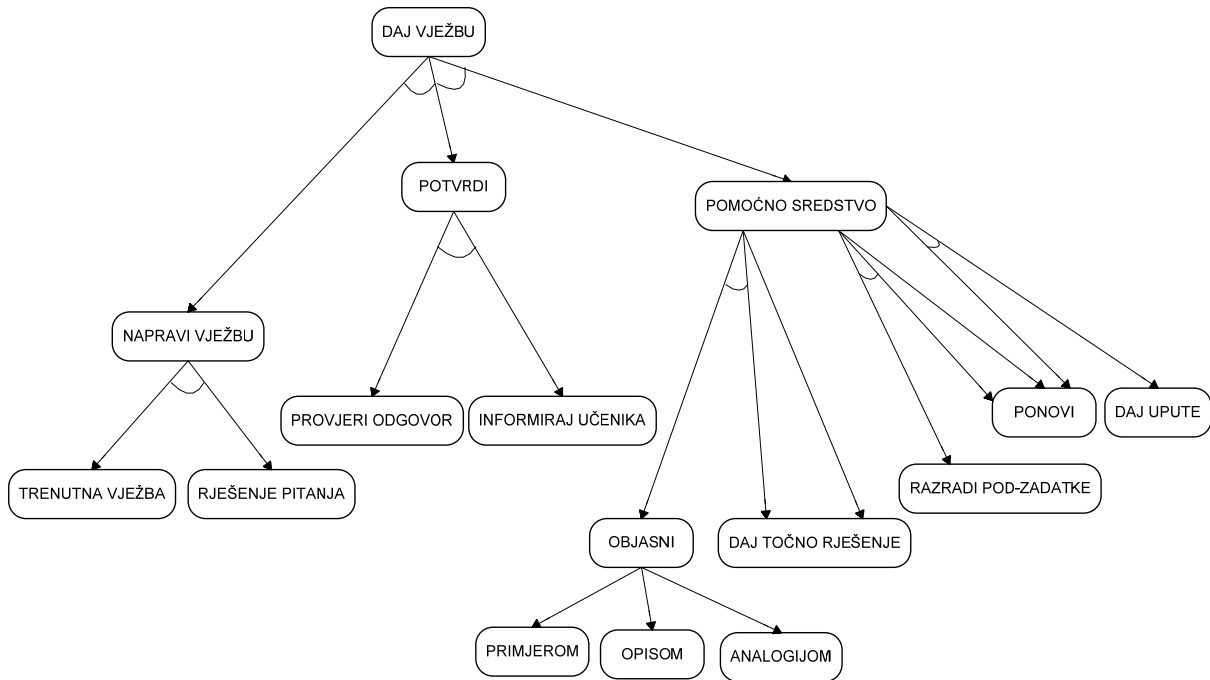
Pedagoška komponenta (*eng. pedagogical component*) se sastoji od dva dijela od kojih svaki ima generičku jezgru i može se proširiti specifičnim znanjem (obrazovni zadaci i metode, pravila poučavanja). U dodatku, pedagoška komponenta sadrži uređivač pravila poučavanja koji učiteljima dopušta mijenjanje i dodjeljivanje novih pravila (Vassileva, 1997).

## OBRAZOVNI ZADACI I METODE

Obrazovni zadaci i metode (*eng. instructional tasks and methods*) čine prvi dio pedagoške komponente, sadrže prikaz obrazovnih zadataka i njihove dekompozicije na pod-zadatke na temelju različitih obrazovnih metoda, kao u GTE-u. Kao i struktura domene, takve dekompozicije obrazovnih zadataka se mogu prikazati I-ILI grafovima, međutim ovdje čvorovi predstavljaju zadatke, a veze predstavljaju metode dekomponiranja zadatka. *I-veze* predstavljaju povezanost pod-zadataka određenog zadatka u skladu sa određenom metodom dekomponiranja zadatka. *ILI-veze* odgovaraju alternativnim metodama dekomponiranja zadatka.

Primjerice, slika 4.3 prikazuje generički zadatak „DAJ\_VJEŽBU“ koji se može dekomponirati na niz sljedećih pod-zadataka: „NAPRAVI\_VJEŽBU“, „POTVRDI“, „POMOĆNO\_SREDSTVO“. Pod-zadatak „POMOĆNO\_SREDSTVO“ se može dekomponirati na različite načine u skladu sa različitim metodama:

- „OBJASNI“ i „DAJ\_TOČNO\_RJEŠENJE“,
- „DAJ\_TOČNO\_RJEŠENJE“,
- „RAŽRADI\_POD-ZADATKE“ i „PONOVI“,
- „PONOVI“,
- „PONOVI“ i „DAJ\_UPUTE“.



Slika 4.3 Primjer hijerarhije zadatka (modificirano prema (Vassileva, 1998))

Cilj hijerarhije obrazovnih zadataka je omogućiti planiranje redoslijeda nastavnih materijala usmjerenih na dani koncept plana sadržaja, tj. plan kao pedagoški značajan način prikaza određenog koncepta. Stoga će se ovakav plan nazivati *plan prikaza* (eng. *presentation plan*), a proces *planiranje prikaza* (eng. *presentation planning*). Zadaci i metode mogu biti generički, ali kao što je opisano, što su pod-zadaci dublje u hijerarhiji dekompozicije zadataka, to su ovisniji o predmetu. Za oblikovanje je dan posebni uređivač u komponenti *modul oblikovanja* (eng. *authoring module*) kako bi se predefinirani skup generičkih obrazovnih zadataka i metoda sustava mogao proširiti općenitijima vezanim uz predmet (Vassileva, 1997).

## PRAVILA POUČAVANJA

Skup pravila poučavanja (eng. *teaching rules*) čini drugi dio pedagoške komponente. Pravila poučavanja provode odabir plana sadržaja i plana prikaza. Većina pravila poučavanja je generička. Pravila poučavanja se mogu svrstati u sljedeće kategorije (Vassileva, 1997):

- diskurzivna pravila (eng. *discourse rules*)
- pravila odabiranja strategije (eng. *strategy-selection rules*)
- pravila odabiranja metode (eng. *method-selection rules*)
- pravila odabiranja nastavnog materijala (eng. *teaching materials selection rules*)

**Diskurzivna pravila** poučavanja provode odabir plana pri planiranju ili ponovnom planiranju na razini sadržaja (strukture domene). Određuju kriterije za odabir plan kada postoji više alternativnih planova (primjerice s obzirom na vrstu semantičkih veza, kada dopustiti izmjenu za slijeđenje drukčije vrste veze) i kako slijediti plan. Također, ako je struktura domene organizirana na osnovu različitih aspekata, definiraju koje aspekte pokriti i kako ih kombinirati.

Neke diskurzivna pravila navode da je za praćenje plana u slučaju inteligentnog učenika prikladna dedukcija (eng. *top - down*) u skladu sa apstrakcijom veza (od općih prema specifičnim), a u slučaju ne tako inteligentnog ili pouzdanog učenika je prikladna indukcija

(eng. *bottom - up*). Ako je potrebno ponovno planiranje na razini koncepta, diskurzivna pravila odabiru hoće li doći do lokalnog mijenjanja ili do globalne promjene plana.

**Pravila odabiranja strategije** definiraju kako odabrati strategiju poučavanja prije izvođenja plana. Strategija poučavanja definira općenite principe poučavanja, primjerice tko ima inicijativu u odlučivanju što učiniti sljedeće, učenik ili sustav. Razlikujemo dvije glavne strategije poučavanja, strukturiranu (eng. *structured teaching strategy*) i nestrukturiranu strategiju poučavanja (eng. *unstructured teaching strategy*).

Kod *strukturirane strategije poučavanja*, inicijativa je u rukama sustava koji odabire koji će se koncepti sljedeći poučavati i kako (tj. s kojim obrazovnim zadacima).

*Nestrukturirana strategija poučavanja* izbor sljedećeg koncepta ostavlja učeniku. Prikaz strukture domene na ekranu i označavanje konceptata „spremnih za poučavanje“ (oni koncepti čiji se preduvjeti u planu smatraju učeniku poznatim) može učeniku pomoći u navigaciji strukturom domene. Učenik može izabrati i obrazovni zadatak i metodu za trenutni koncept iz grafičkog prikaza hijerarhije zadataka. Neka pravila odabiranja strategije primjerice navode da je za motiviranog i uspjehom vođenog učenika prikladna nestrukturirana strategija, dok je za nesigurnog i ne-samopouzdanog učenika prikladnija strukturirana strategija.

I diskurzivna pravila i pravila odabiranja strategije uzimaju u obzir podatke iz modela učenika, kao i vanjske čimbenike (kao što je npr. vrijeme).

**Pravila odabiranja metode.** Obično postoji od tri do osam alternativa dekomponiranja zadatka i metoda za svaki obrazovni zadatak (Van Marcke, prema Vassileva, 1997). Pravila odabiranja metoda uzimaju u obzir povijest korištenih obrazovnih zadataka i nastavnih materijala te učenikove osobne poteze i preferencije (iz modela učenika) za odabir obrazovnih zadataka za trenutni koncept i za odabir metode dekomponiranja zadataka. Ovo se provodi u fazi planiranja prikaza. Pravila odabiranja metode rješavaju problem koji se javlja u GTE-u definicijom uvjeta primjene metode dekomponiranja zadatka, tj. kako među mogućim alternativnim metodama odabrati metodu za dani zadatak.

U pedagoškoj literaturi su pronađene četiri alternativne metode za poučavanja koncepta koje dekomponiraju glavni zadatak na pod-zadatke.

*Metoda hijerarhije* (eng. *hierarchical method*) poučava sljedećim redoslijedom pod-zadataka: uvedi, objasni, potkrijepi primjerom, daj vježbu i daj test.

*Metoda naprednog organizatora* (eng. *advanced organizer method*) dekomponira zadatak sa dodatnim prvim pod-zadatkom koji učeniku eksplicitno predstavlja trenutni cilj poučavanja i plan pod-zadatka koji će se izvršiti i objektivnu izjavu što se očekuje postizanjem trenutnog cilja, tj. koja je važnost poučavanja trenutnog koncepta za globalni cilj.

Prvi pod-zadatak *metode osnovnog koncepta* (eng. *basic concept method*) je postaviti problem (vježba) čija rješenja zahtijevaju znanje o ciljnom konceptu. Obzirom da učeniku nedostaje znanje vezano uz baš taj koncept koji će se poučavati, od učenika se ne očekuje da riješi problem. Međutim, pokušaji rješavanja problema ga pripremaju da razumije potrebu za uvođenjem novog koncepta. Tada se uvodi novi koncept popraćen objašnjenjima i primjerima rješavanja takve vrste problema, potom vježba i na kraju test.

*Metoda otkrića* (eng. *discovery method*) uključuje prikaz problema motivacije, analiziranje problema i puštanje učenika da sam riješi problem, nadajući se da će otkriti koncept pri rješavanju problema.

Sljedeći Einsiedler-a prema (Vassileva, 1997) definiramo skup pravila:

- metoda osnovnog koncepta se preferira za uspješne, ali ne baš koncentrirane i motivirane učenike
- metoda naprednog organizatora se preferira za uspješne, ali ne baš koncentrirane i samopouzdanu učenike
- metoda hijerarhije se preferira za koncentrirane učenike bez dosadašnjeg uspjeha
- metoda otkrića se preferira za pouzdane i koncentrirane učenike

**Pravila odabiranja nastavnog materijala.** Kada se odabere trenutni obrazovni pod-zadatak i dekomponira na primitivne pod-zadatke, pravila odabiranja nastavnog materijala odlučuju o načinu biranja nastavnih sadržaja odgovarajućih vrsta medija (tekst, grafički zapis, animacija, video, i sl.). Kako bi među brojnim nastavnim materijalima za odabrani primitivni pod-zadatak odabrali one nastavne materijale one određene vrste medija koju učenik preferira uzimaju u obzir model učenikovih osobnih poteza i preferencija. Međutim, mogu dati prvenstvo određenoj vrsti medija koju učitelj preferira.

## UREĐIVAČ PRAVILA POUČAVANJA

Uređivač pravila poučavanja (*eng. teaching rules editor*) omogućava učitelju da definira vlastita pravila za odabiranje strategije poučavanja, metode i nastavnog materijala što se postiže dodjeljivanjem uvjeta za primjenu pravila (varijable modela učenika) i učinka (odluke izbora). Jasno je da skup pravila poučavanja ima iznimno važnu ulogu u primjeni DCG-a. Međutim, sam uređivač pravila poučavanja ne rješava problem kreiranja pravila. Kako postići takva pravila? Tri su moguća pristupa, teorijski, osobni i empirijski pristup (Vassileva, 1997).

*Teorijski pristup (eng. theory-based)* sastavlja pravila vodeći se postojećim didaktičkim teorijama – trenutno rješenje. Nedostatak ovog pristupa je taj što je većina didaktičkih teorija prilično općenita i ne formulira precizna pravila. Dizajner (autor, učitelj) mora interpretirati općenite smjernice opisane u tim teorijama kako bi došli do konkretnih pravila koja mogu voditi akciju u specifičnoj situaciji. Ova interpretacija je uvijek subjektivna i podložna je kritikama.

*Osobni pristup (eng. personal-based)* uključuje intervjuiranje učitelja ili implementiranje pravila izravno od strane učitelja. Ovaj pristup zahtijeva da su učitelji u mogućnosti raščlaniti čimbenike koji utječu na njihovu odluku, što često nije slučaj. Obzirom da postoje dokazi da se ljudi odnose prema svojim odlukama na različit način nego što ih zapravo donesu (jer su usmjereni na „post-mortem“ logička objašnjenja svojih akcija), ova metoda može dovesti do nevažećih pravila.

*Empirijski pristup (eng. empirically-based)* uključuje analiziranje protokola stvarnih individualnih sesija poučavanja, identificiranje slučajeva i primjenjivanje tehnika strojnog učenja za generiranje stabala odluke i pravila. Ovaj pristup bi najvjerojatnije dao najpouzdanije rezultate, no obzirom da zahtijeva napredne alate strojnog učenja i mnoštvo empirijskih podataka, teško ga je realizirati.

### 4.2.4 Generator kolegija

Generator kolegija (*eng. course generator*) je komponenta koja kreira kolegij, ostvaruje komunikaciju sa učenikom i održava model učenika. Generator kolegija se sastoji od sljedećih komponenti: planer kolegija i izvršitelj kolegija (Vassileva, 1997).



## PLANER KOLEGIJA

Planer kolegija (*eng. course planner*) je program planiranja I-ILI grafa koji se može pozvati sa dvije namjene:

- generiranje plana sadržaja (koncepti koji će se prikazati u kolegiju)
- kreiranje plana prikaza (redoslijeda kolegija) za poučavanje ciljnog koncepta

Učitelj poziva planera kolegija za određenog učenika i dodjeljuje ciljeve poučavanja za kolegij, daje skup aspekata kojeg plan mora pokriti, vrste veza obzirom na plan i maksimalnu dubinu pretraživanja grafa. Ako postoje diskurzivna pravila koja dodjeljuju ove parametre određenom cilju poučavanja, zadatak učitelja se svodi na odabiranje ciljnih koncepata za poučavanje. Planer se aktivira za kreiranje plana kolegija. Algoritam planiranja je modifikacija AO\* algoritma za pretraživanje grafa (Nilsson, 1980). Funkcija optimizacije  $h$  se može odabrati tako ostvari različite kriterije za optimalnost, (tj. za odabir plana), npr. najkraći plan, plan koji izbjegava određeni koncept, plan određene topologije i slično. Odabir funkcije  $h$  se provodi diskurzivnim pravilima. Rješenja I-ILI grafa počinju od koncepata koji su učeniku poznati (imaju visoku vjerojatnost znanja u modelu znanja učenika) i vode do ciljnog koncepta. Plan nameće samo parcijalno uređenje koraka rješavanja problema. Završno uređivanje pod-ciljeva se provodi u vrijeme pokretanja na osnovu specifičnih diskurzivnih pravila određene domene i u skladu sa odabranom strategijom poučavanja.

## IZVRŠITELJ KOLEGIJA

Izvršitelj kolegija (*eng. course executor*) prima plan od planera i generira kolegij koji se temelji na odabiru i prikazu odgovarajućih nastavnih materijala. Odabire se glavna strategija poučavanja (strukturirana ili nestrukturirana) za poučavanje provjeravanjem pravila za odabiranje strategije.

Ako je odabrana nestrukturirana strategija, učenik iz grafičkog prikaza plana mora izabrati sljedeći koncept koji želi učiti i obrazovnu metodu. Ako je odabrana strukturirana strategija, izvršitelj se konzultira sa diskurzivnim pravilima i odabire trenutni koncept ili vezu za poučavanje, onda se konzultira sa pravilima za odabiranje metode i odabire metodu. Tada poziva planera za kreiranje plana obrazovnih pod-zadataka potrebnih za implementiranje odabrane metode, potom se konzultiraju pravila odabiranja nastavnih materijala te se odabiru odgovarajući nastavni materijali.

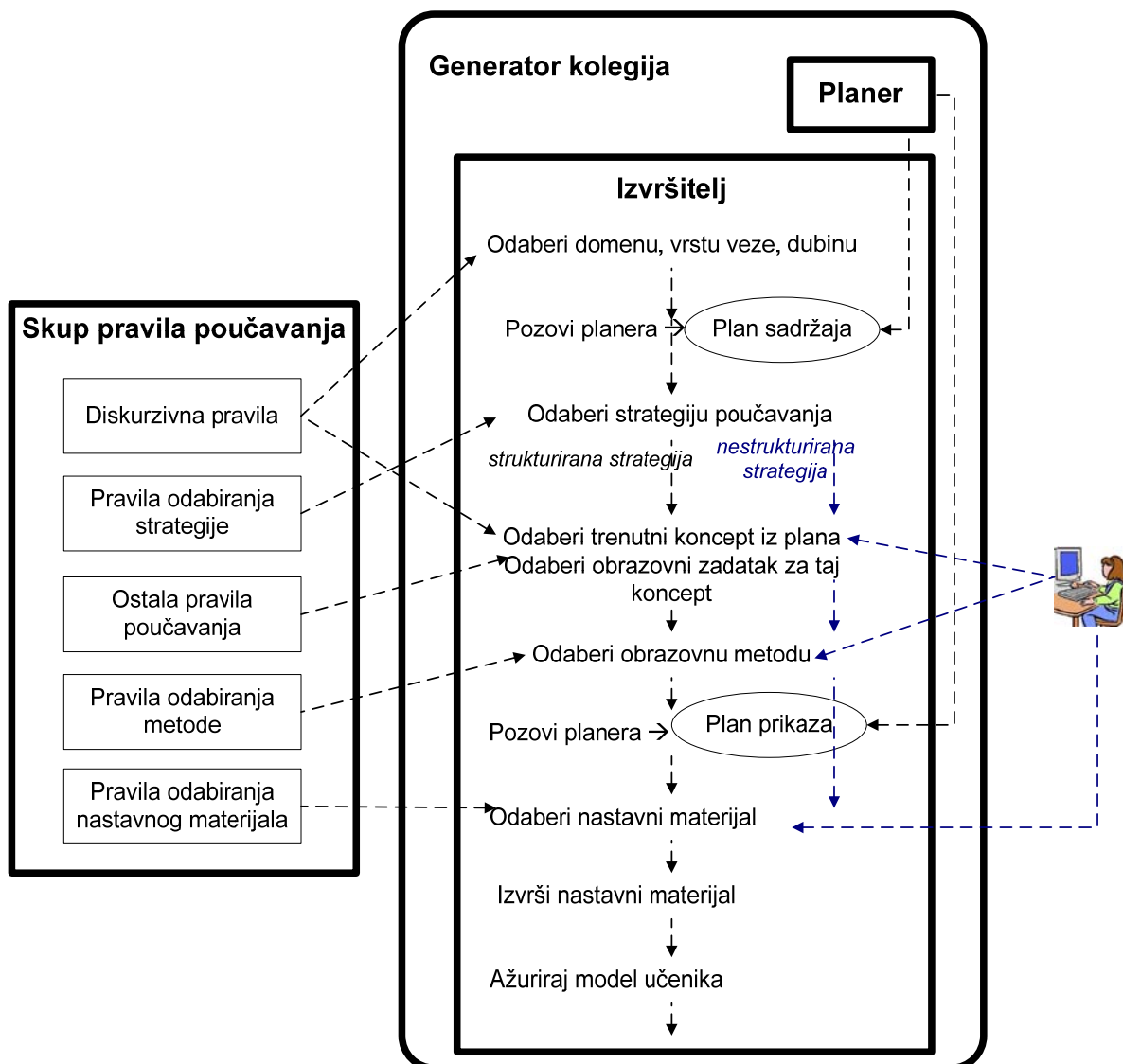
Odabrani nastavni materijal se prezentira u skladu sa pod-zadatkom poučavanja, potom se izvršava sljedeći pod-zadatak itd., sve dok se ne provede testiranje nastavnog materijala koje provjerava usvojenost koncepta i tada se model učenikovog znanja ažurira na osnovu uvjetnih vjerojatnosti nastavnih materijala.

Tijekom poučavanja, postoji aktivno polje na ekranu (botun) koji učeniku dopušta mijenjanje vrijednosti svojih osobnih preferencija u modelu učenika. Pri svakom odabiru obrazovne metode ili nastavnog materijala za sljedeći koncept u obzir se uzima ažurirano stanje modela učenika.

Tablica 4.2 Razlike između plana sadržaja i plana zadatka (modificirano prema (Vassileva, 1998))

	Plan sadržaja	Plan prikaza
<b>I-ILI graf predstavlja:</b>	Struktura domene	Obrazovni zadaci i metode
<b>Čvorovi:</b>	Koncepti	Obrazovni zadaci
<b>Veze:</b>	Semantičke veze među konceptima	Metoda dekomponiranja zadatka
<b>Odabir provode:</b>	Diskurzivna pravila, pravila odabiranja strategije	Pravila odabiranja metode

Na slici 4.4 je prikazan proces dinamičkog generiranja courseware-a.



Slika 4.4 Generiranje i izvršavanje kolegija (modificirano prema (Vassileva, 1997))

Neovisno o odabranoj strategiji, može se dogoditi da učenik ne usvoji dani koncept unutar predviđenog vremena. Simptom toga je nedovoljna vjerojatnost znanja tog koncepta u modelu učenika („dovoljno“ je vjerojatnostni prag kojeg definira autor). U tom slučaju, izvršitelj poziva planera da kreira novi plan za koncept zaobilaženjem teškog koncepta.

Postoje dvije glavne vrste ponovnog planiranja: plan lokalnog popravka i globalno ponovno planiranje.

- *Plan lokanog popravka* (eng. *local plan repair*) mijenja samo one dijelove plana koji se odnose na trenutni cilj. Na ovaj način, sustav pokušava pronaći alternativni način za poučavanje teškog koncepta bez mijenjanja cjelokupnog plana.
- *Globalno ponovno planiranje* (eng. *global re-planning*) traži alternativni plan za glavni cilj poučavanja.

Diskurzivna pravila određuju koja će se vrsta ponovnog planiranja izabrati.

## 4.2.5 Modul oblikovanja

Komponenta oblikovanja (*eng. authoring module*) se sastoji od uređivača nastavnih materijala, uređivača strukture domene i uređivača obrazovnih zadataka i metoda (Vassileva, 1997).

### UREĐIVAČ NASTAVNIH MATERIJALA

Uređivač nastavnih materijala (*eng. teaching materials editor*) je alat koji omogućava „pakiranje“, tj. prikaz nastavnih materijala tako da se sustav može koristiti nastavnim materijalima kreiranim bilo kojim autorskim alatom za izradu multimedijalnih materijala. Gotovi CAL materijali, kolegiji, video - zapisi i grafički zapisi se mogu ponovno koristiti. Svakom nastavnom materijalu je dano jedinstveno ime i povezano je sa jednim konceptom ili vezom iz strukture domene. Da bi se uključili u bazu podataka, potrebno je klasificirati nastavne materijale obzirom na pedagošku vrstu i medij te dodijeliti predviđeno vrijeme za taj nastavni materijal.

Oni nastavni materijali s kojima učenici surađuju se moraju uključiti u bazu podataka kao skup „čestica“: ovo je lista kazaljki koja odgovara „tijelu“ nastavnih materijala (pitanje, problem i sl.) za točni odgovor, praktične upute, objašnjenje ili dekomponiranje rješenja na korake. Svi ovi materijali su pojedinačno dostupni pomoću obrazovnih pod-zadataka. Ukoliko nisu prikazane sve gore navedene „čestice“, odgovarajući nastavni materijali se još uvijek mogu koristiti za glavni zadatak, no neke metode dekomponiranja zadatka neće biti primjenjive. Primjerice, pretpostavimo da u bazi podataka za određeni koncept A postoje dva nastavna materijala s pedagoškom karakteristikom „vježba“: koji su prikazani kao lista kazaljki sljedećih „čestica“:

$E_{X1} \rightarrow (*navod\_problema, *točan\_odgovor, *uputa)$

$E_{X2} \rightarrow (*navod\_problema, *točan\_odgovor, *uputa, *pohranjeni\_koraci\_rješenja, *objašnjenja\_rješenja)$

Ako je za implementiranje zadatka „DAJ\_VJEŽBU“ za koncept A odabran  $E_{X1}$ , dostupne su samo tri od mogućih pet alternativnih metoda dekomponiranja zadatka za pod-zadatak „POMOĆNO\_SREDSTVO“. Ostale dvije metode će biti primjenjivanje sa  $E_{X2}$  jer ima čestice potrebne za implementiranje svih njihovih pod-zadataka.

Kako bi osposobili sustav za izvršavanje obrazovnih pod-zadataka koji daju povratnu informaciju o učenikovoju pogrešci, autor za svaku vježbu mora kreirati pridruženi pomoćni materijal. To može biti uputa, objašnjenje, koraci rješenja, usmjereno pitanje itd. koje će se prikazati učeniku pomoću obrazovnih pod-zadataka, pri čemu će poredak biti propisan planom zadatka.

Trebao bi se kreirati barem jedan „atomski“ test za svaki čvor i vezu, tako da sustav može suditi o učenikovom uspjehu odnosno neuspjehu ovisno o tome znali li pridruženi koncept. Kako bi se sustavu osigurali načine vrednovanja razine znanja svakog čvora (veze) uključenog u „atomski“ test, autor mora definirati vektore vjerojatnosti, to jest, dva vektora koji sadrže uvjetne vjerojatnosti učenikovog znanja o svakom uključenom čvoru (vezi), što ovisi o (ne)točnim odgovorima na dani „atomski“ test. Iako nema jamstva da je dana vjerojatnost adekvatna, iskustvo pokazuje da autoru nije problem dati aproksimativnu procjenu vjerojatnosti.

## UREĐIVAČ STRUKTURE DOMENE

Uređivač strukture domene (*eng. domain structure editor*) je grafički uređivač koji omogućava razvijanje, proširenje i mijenjanje strukture domene. Podržava kreiranje, brisanje i izmjenu aspekata. Za odabrani aspekt podržava umetanje, brisanje, premještanje, imenovanje i preimenovanje prikazanih čvorova te umetanje, brisanje i povezivanje veza, prikaz različitih semantičkih veza drukčijim bojama, pregled postojećih nastavnih materijala u bazi podataka i pridruživanje tih materijala konceptima i vezama iz strukture domene. Ovaj uređivač pruža mogućnost pridruživanja uvjetnih vjerojatnosti kao vrstu posebne veze između koncepta neovisnu o vezama koje označavaju relacije među konceptima.

## UREĐIVAČ OBRAZOVNIH ZADATAKA I METODA

Uređivač obrazovnih zadataka i metoda (*eng. instructional tasks and methods editor*) je sličan uređivaču strukture domene. Dopušta kreiranje, brisanje i mijenjanje strukture obrazovnih zadataka. Alternativne metode dekomponiranja zadatka su predstavljene povezivanjem čvorova (zadatka) sa lukovima (*eng. arcs*) različitih boja, debljine i uzorka. Svakom listu (čvor koji se ne može dekomponirati) pod-zadatka je priskrbljena lista odgovarajućih pedagoških vrsta nastavnih materijala koje se može prikazati. Većina hijerarhija zadataka i metoda dekomponiranja je generička i unaprijed definirana. Uređivač dopušta autoru definiranje predmeta obrazovnih zadataka.

## 4.3 Dinamičko generiranje i oblikovanje kolegija

Za generiranje kolegija je odgovoran planer (*eng. planner*), a za oblikovanje je odgovoran autorski alat (*eng. authoring tool*) (Vassileva, 1994).

### GENERIRANJE PLANA

Planer automatski generira plan kolegija na temelju strukture domene. Ukoliko ne postoje odgovarajuća diskurzivna pravila, glavni unos za kreiranje plana su ciljevi poučavanja dodijeljeni od strane učitelja. Učitelj može dodijeliti ciljeve poučavanja (misli se na čvor ili vezu strukture domene) i postaviti zahtjeve za plan kolegija koji mogu biti:

*Dimenzije i razine koje kolegij mora istražiti.* Primjerice, „pouči komponentu  $X$  do najniže razine pod-komponentata“.

*Čvorovi ili veze koje treba uključiti.* Primjerice, ukoliko učitelj želi da kolegij uz objašnjenja funkcije  $f$  i pregled pod-funkcija sadrži i pregled kako se ove funkcije implementiraju u komponente, objedinit će funkcionalnu i strukturalnu dimenziju veze.

*Odabiranje vrste „atomske“ prezentacija i težine „atomske“ testova.* Ukoliko ništa ne dodijeli, sustav će dinamički odabrati na osnovu prve razine modela učenika.

*Raspodjela vremena za kolegij.* Ukoliko učitelj odrediti vrstu „atomske“ prezentacija, izračuna se zbroj njihovih vremenskih trajanja te sustav pokušava uključiti što je više moguće koncepta i veza u tom dijelu vremena.

Generator kolegija kreira plan uključivanjem koncepata (veza) koje učenik ne poznaje, ali uvijek nastoji započeti sa poznatim konceptima (vezama), ili barem s onim konceptima (vezama) koji imaju dovoljno visoku vjerojatnost znanja.

## **OBLIKOVANJE KOLEGIJA**

Autorski alati (*eng. authoring tool*) omogućuju jednostavno kreiranje i ažuriranje baze podataka. Odlučuje se između autora i učitelja tko će koristiti gotov sustav u svom području i uskladiti ga sa svojim zahtjevima.

Od autora se očekuje da kreira prvu strukturu domene za danu domenu, uz pomoć uređivača strukture domene koji vizualizira čvorove i veze među njima i daje znak spremnosti za prihvatanje informacije koju autor mora unijeti. Autor mora kreirati i bazu podataka nastavnih materijala koristeći se komercijalnim autorskim alatom i svakom od njih mora dodati parametre koji govore o vrsti i raspodjeli vremena. Svaki podatak za koji se očekuje da će se mijenjati, mora se parametrizirati. Realne vrijednosti se čuvaju u strukturi domene: zamijenit će parametre u odgovarajućim nastavnim materijalima u fazi izvršavanja kolegija. Pri ažuriranju tih vrijednosti, biti će potrebno mijenjati vrijednosti parametara pripadajućih koncepata odnosno veza. Sustav pomaže autoru da prenosi promjene parametara u strukturu domene.

Ako autor želi postići reaktivnost sustava, to jest omogućiti izravnu povratnu informaciju o pogrešci učenika, mora za svaki „atomske“ test kreirati dopunski materijal. Ovo se može ostvariti na različite načine, primjerice, prozor sa objašnjenjem pogreške i po mogućnosti pružanjem točnog odgovora. Obzirom da se ne cilja na istinski inteligentni sustav, nije potrebna nikakva dijagnoza razloga javljanja pogreške. Umjesto poučavanja navođenjem učenikovih zabluda, sustav pokušava poučavati alternativnim prikazima.

## **4.4 Dimenzije tehnike opisa domene**

Struktura domene predlaže homogenu mrežu koncepata i veza različitih vrsta među njima. Veze mogu imati različito semantičko značenje, npr. agregacija, apstrakcija, analogija. Kako bi generirali značajan kolegij za poučavanje danog koncepta, planeru kolegija je potrebno znanje o semantici veza koje se odnose na koncepte koji se moraju koristiti (Vassileva, 1994).

Obzirom da postoje različiti načini poučavanja jednog koncepta, odluka kojim se načinom voditi je dio cilja poučavanja pa semantiku veza mora dodijeliti učitelj. Time je otežan posao učitelja jer bi trebao znati sve koncepte koji se koriste i sve različite veze među njima. Potreban je način sistematiziranja eventualnih veza između koncepata.

Nemoguće je opisati složene tehničke sustave bez uzimanja u obzir strukturirani prikaz koja dopušta različite aspekte i razine apstrakcije u opisu. Primjerice, u dizajniranju čipova, koristi se standardna tehnika koja razlikuje tri dimenzije, tzv. konstrukcijski, funkcionalni i geometrijski aspekt opisa čipa pri čemu svaki od ova tri aspekta ima različitu razinu apstrakcije. Svaka dimenzija omogućava nezavisan opis čipa u skladu sa dimenzijom.

Za dani cilj poučavanja, često je dovoljno uzeti u obzir samo jednu od ovih dimenzija i to na jednoj razini apstrakcije. Međutim, ponekad se uređaj mora opisati u odnosu na nekoliko različitih dimenzija, stoga bi prikaz trebao dopustiti izmjenu sa jedne dimenzije na drugu, po

možnosti na različitoj razini detalja. Primjerice, pri objašnjavanju strukture mehanizma tostera, može se objasniti i funkcija termostatskog uređaja.

Kako bi dopustili razlikovanje različitih aspekata u opisu uređaja ili sustava, uvodi se pojam dimenzije na razini koncepta. Uređaj je opisan pomoću čvorova i veza. Veze mogu biti jednodimenzionalne i na jednoj razini (reflektiraju strukturu u opisu). Primjerice, u strukturalnoj dimenziji, veze između čvorova odgovaraju komponentama i njihovim podkomponentama, u funkcionalnoj dimenziji se odnose na funkcije i pod-funkcije. Mogu postojati dvije različite razine veza, koje se odnose na apstrakciju ili na agregaciju koncepta. Jednodimenzionalne veze su obično hijerarhijske. Postoje ukršteno-dimenzionalne veze koje imaju određeniju semantiku.

Primjerice, veza sa semantikom „*je implementirana pomoću*“ može povezati čvor odgovarajuće funkcije sa čvorovima odgovarajućih komponenti strukturalne dimenzije koje implementiraju funkciju.

Parametri strukture sadrže one informacije o uređaju koje su predmet mijenjanja tijekom vremena. Kako bi planer kolegija odlučio hoće li čvor prikazati učeniku (u ovisnosti o ciljevima poučavanja), koristi se dijelovima prikaza strukture kao što su dimenzija, razina, izvor i određište. Izvršitelj učenja se koristi dijelovima prikaz i test kako bi odabrao „atomske“ prezentacije, testove odgovarajućeg trajanja i težine za učenika.

## 4.5 Prednosti DCG-a

DCG pristup pruža alternativu tradicionalnom pristupu oblikovanja računalom potpomognute nastave. Njegove glavne prednosti su (Vassileva, 1995):

- *Fleksibilnost ciljeva kolegija* omogućuje više-aspektna organizacija koncepta i mogućnost definiranja različitih vrsta semantičkih veza među kojima sustav može odlučiti kako će planirati kolegij za dani cilj, na optimalan način pomoću raznih diskurzivnih pravila.
- *Individualizacija nastave*. Poučavanje složenih tehničkih sustava zahtijeva uzimanje u obzir da učenici imaju različite pozadine, obrazovanje, razinu znanja, motivaciju, inteligenciju, pouzdanje i neovisnost. Stalnim nadziranjem učenikovog ponašanja i održavanjem modela učenika, DCG može pronaći alternativne planove kolegija, metode poučavanja i nastavne materijale koji se dinamički usklađuju sa učenikovim potrebama.
- *Mogućnost dodjeljivanja i mijenjanja strategija poučavanja*. Ljudski učitelj mora imati jasne mehanizme uspoređivanja funkcionalnosti sustava. Sustav se može promatrati kao agenta poučavanja plana, poput kurikulumu, koji izvodi obrazovne zadatke pomoću specifičnih metoda. Ovaj agent se može poučiti pomoću dodavanja pravila poučavanja u kojim slučajevima odabrat koji plan, metodu i nastavne materijale.
- *Mogućnost jednostavnog oblikovanja*. Ponovna upotreba razvijenog courseware-a i tehničke dokumentacije kao osnova za razvijanje nastavnih materijala. Ovo je sadržano odvajanjem strukture domene od trenutnih nastavnih materijala što dopušta izravno ažuriranje i mijenjanje bez mijenjanja strukture domene.

Oblikovanje kolegija je podijeljeno između autora (projektant baze podataka određene domene) i učitelja čime se učitelje aktivno uključuje u kreiranje kolegija bez previše truda i

posebnog znanja o oblikovanju i autorskom alatu i pomaže u sveukupnom odobravanju sustava. Aktualan proces oblikovanja je prebačen na viši nivo: eksplicitno predstavljanje strukture domene i zadataka poučavanja.

Navedene su sljedeće osnovne primjene DCG-a (Vassileva, 1997):

- *Podrška predavanja (eng. lecture support), obrazovanje na daljinu (eng. distance education) i trajno usavršavanje (eng. continuous education).* Kolegiji generirani DCG-om se mogu koristiti kao dodatni nastavni materijali (kao interaktivna skripta) za podršku predavanja (redovnu ili povremenu) na sveučilištu. Interaktivni kolegiji na Web-u prateći ponuđena predavanja na sveučilištu pružaju neprestano ažuriranje znanja.
- *Ponovna upotrebljivost i dijeljenje domene.* Distribuirana arhitektura DCG-a dopušta autorima suradnju u uređivanju struktura domena i povezanih nastavnih materijala koncepata. Dopušta i ponovno korištenje nastavnih materijala i struktura domena. Mogu se razviti biblioteke često korištenih „osnovnih“ koncepata. Domenu prikazanu u DCG-u mogu dijeliti predavači koji poučavaju isti predmet na različitim sveučilištima, pri čemu svaki predavač može uvesti promjene ili proširenje i koristiti se strukturama i materijalima re kombiniranjem u skladu sa vlastitim osobnim pogledima za pripremanje kolegija predavanja (nisu utemeljena na računalu).
- *Učenici kao autori.* Moderne teorije poučavanja ističu pozitivne ishode učenikovog osjećaja „posjedovanja“ problema te učinke objašnjenja i samo-objašnjenja razvoja reflektivnih vještina i vlastite strukture znanja (koja nije nužno točna). Zbog ovog je razloga učenicima često prepušteno da sami planiraju lekciju, primjerice organiziranjem predavanja kao seminara što im omogućava da kreiraju vlastiti nezavisni pogled domene i sposobnost traženja nove informacije, razvijanje strukture domene i integriranje u već postojeće. DCG kao autorski alat se može koristiti za iznošenje ovakve vrste projekta. Primjerice, učeniku ili grupi se može dodijeliti zadatak oblikovanja određene teme (pod-domena). Učenik odnosno grupa mora pregledati literaturu, otkriti važne koncepte i veze te kreirati strukturu domene, kreirati ili pronaći povezane materijale na Web-u te ih povezati sa konceptima strukture domene. Strukturu i materijale će razmatrati i kritizirati predavač i razred, a ovako izrađenom domenom se DCG može kasnije koristiti za automatsko generiranje obrazovnih kolegija na tu temu.

## 4.6 Implementacija DCG-a na WWW-u

Kao što je spomenuto ranije, razvijen je alat za oblikovanje prilagodljivih kolegija, nazvan *dinamičko generiranje courseware-a* (DCG) (Vassileva, 1994). Generira individualne kolegije u skladu sa učiteljevim ciljevima i prethodnim znanjem, dinamički prilagođava kolegij na osnovu učenikovog uspjeha stjecanja znanja i vještina. DCG se pokreće na Web poslužitelju. Učenik od poslužitelja prima individualizirani kolegij usmjeren na određeni cilj, nakon toga je prilagodljivo vođen kolegijem kroz nastavne materijale na Web-u. Za razliku od drugih CAL kolegija na Web-u, kolegij izrađen DCG-om je interaktivan, testira učenikovo znanje i dinamički se prilagođava učenikovom napretku. Autorski alat se može koristiti i za suradničko oblikovanje i učenje (Vassileva i Deters, 1998).

Pojavljivanjem WWW-a, učenicima je moguće osigurati neograničeni pristup nastavnim materijalima. Obzirom da se za stvaranje takvih materijala (html dokumenata) mogu koristiti obični uređivači teksta, samim time je pojednostavljeno oblikovanje. Međutim, učenici se na Web-u suočavaju sa jednim značajnim problemom: moraju izaći na kraj sa velikom količinom materijala i linkovima koji ne moraju biti relevantni za njihove ciljeve učenja. Često se izgube u hiperprostoru, zaborave gdje su stali i izgube usmjerenu perspektivu polja. Postoje različiti pristupi za podršku navigacije korisnika pri učenju sa Web-a. Većina njih koristi pomoćnu pedagošku strukturu domene povezanu vrstama veza *preduvjet* koja prekriva hiperlink strukturu dokumenata. Primjerice, AST koristi *preduvjet* strukturu koncepta za predlaganje koji će se koncepti prvi poučavati s namjerom da se razumije cilj poučavanja koncepta. ELM-ART (eng. *ELM Adaptive Remote Tutor*) obilježava veze trenutnog koncepta sa „spreman za poučavanje“ ili „još nije spreman za poučavanje“. Ovakvi pristupi nude vrstu navigacijske podrške nazvanu *lokalno orijentirana podrška* (Brusilovsky, prema Vassileva i Deters, 1998).

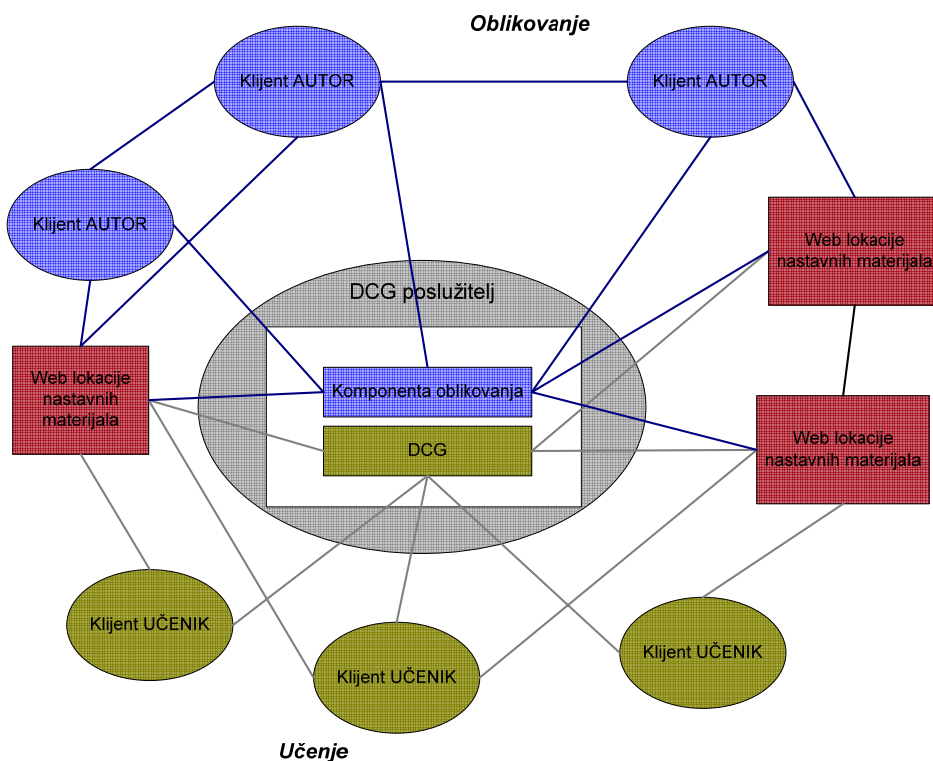
Za učenikove potrebe je prikladnije prilagodljivo vođenje (eng. *adaptive guidance*) kroz materijale od lokalno orijentirane podrške obzirom da provodi usklađeni način poučavanja (usmjeren na ciljeve) koji je prikladan kada je vrijeme ograničeno, a učenici su motivirani (Vassileva i Deters, 1998). CAI (eng. *Computer Aided Instruction*) pruža ovakvu vrstu poučavanja – tradicionalni CAL kolegij je predefiniрани poredak nastavnih materijala (od kojih neki mogu biti interaktivni) usmjeren na postizanje nekih ciljeva poučavanja. CAL kolegij se dizajnira imajući na umu određenu populaciju učenika za koju se smatra da već posjeduje specifično znanje. Ovo rješenje je međutim prilično kruto za prihvaćanje učenika na WWW-u sa šarolikim ciljevima poučavanja, pozadinskim znanjem i različitim načinom stjecanja materijala. Ovaj problem se može riješiti dajući učenicima Web pristup alatu za automatsko generiranje individualiziranih prilagodljivih kolegija.

DCG dopušta:

- Automatsko sastavljanje CAL kolegija sa različitim ciljevima iz pričuve nastavnih materijala
- Kreiranje različitih kolegija za učenike sa različitim znanjem
- Dinamičko mijenjanje kolegija u skladu sa napretkom učenika

Posvojena je klijent - poslužitelj arhitektura za DCG na WWW-u (slika 4.5). DCG je smješten na Web poslužitelju, nudi usluge poučavanja i oblikovanja. Klijent može biti bilo koji Web preglednik. Učenici i autori odnosno učitelji su klijenti i koriste usluge poučavanja i oblikovanja ponuđene od strane DCG poslužitelja. Nastavni materijali su distribuirani na različitim lokacijama na WWW-u.



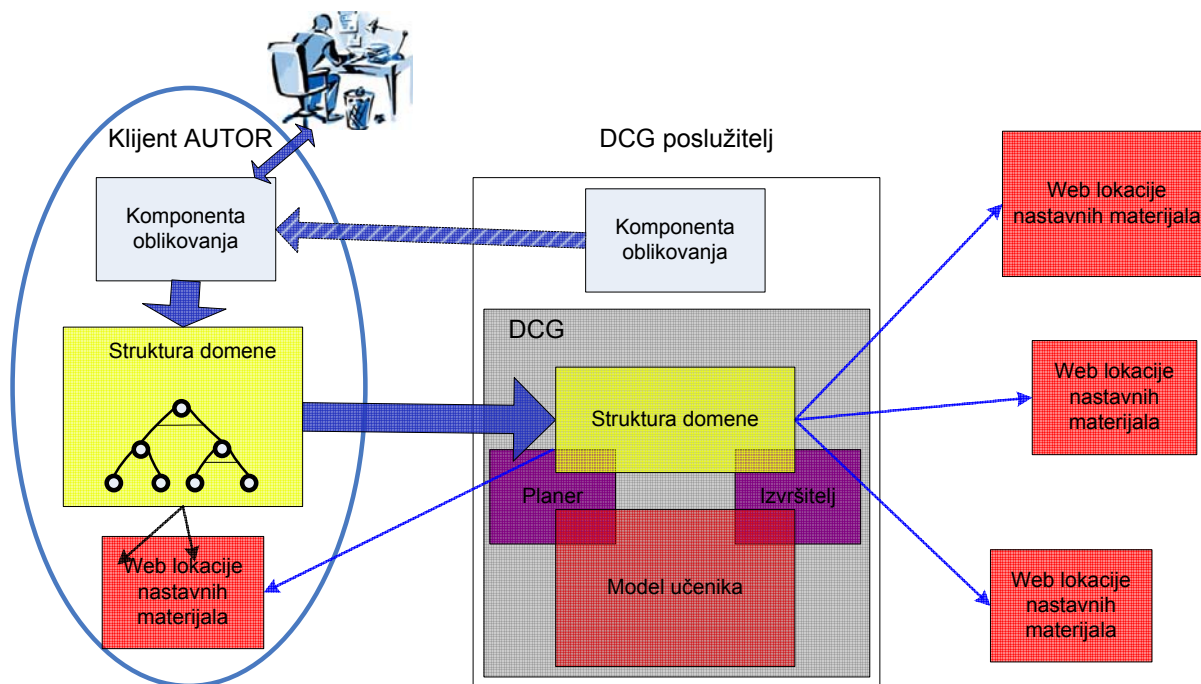


Slika 4.5 Klijent - poslužitelj arhitektura DCG-a (modificirano prema (Vassileva i Deters, 1998))

### OBLIKOVANJE DCG-om na WWW-u

Oblikovanje se sastoji iz dva dijela (vidi sliku 4.6):

- kreiranje ili mijenjanje strukture koncepta dane domene
- osigurava podršku za svaki koncept (vezu) odgovarajućim html dokumentima na Web-u koje se mogu koristiti kao nastavni materijali za taj koncept



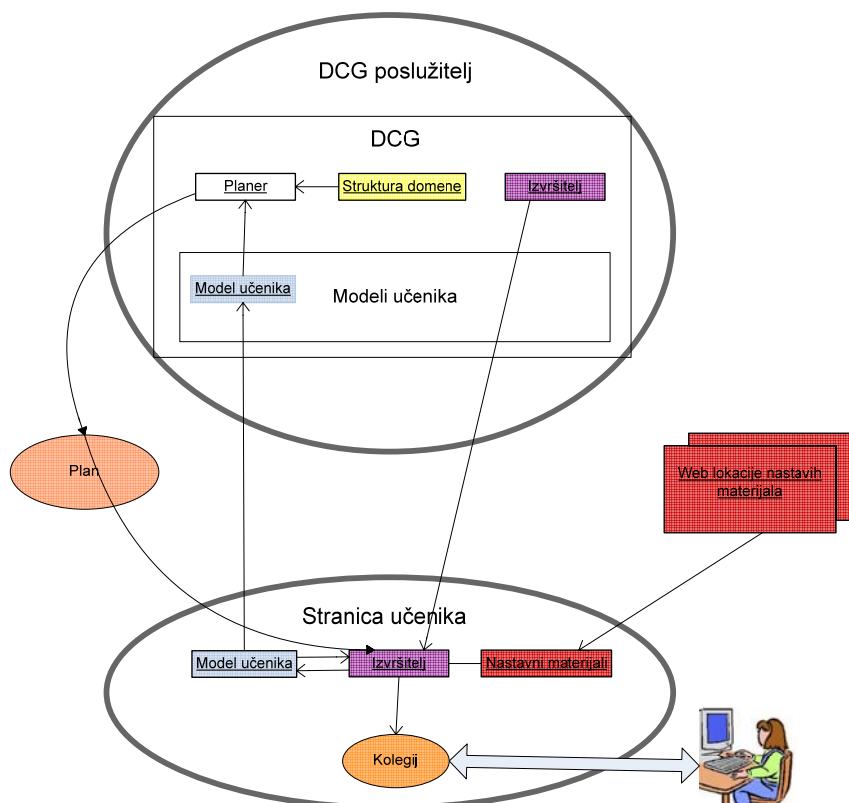
Slika 4.6 Oblikovanje DCG-om na WWW-u (modificirano prema (Vassileva i Deters, 1998))

Specijalni grafički uređivač strukture domene dopušta prikaz koncepata njihovim međusobnim povezivanjem različitih vrsta semantičkih veza. Osiguran je skup standardnih veza (apstrakcija, združivanje, analogija, posljedica), no autor može definirati nove ako vjeruje da su prikladni za strukturiranje domene. Za tu namjenu vezi mora dati ime i definirati grafička svojstva za vizualizaciju relacije u I-ILI grafu koji prikazuje strukturu domene.

Svaki koncept i veza su povezani sa http adresama jednog ili više nastavnih materijala (prikazani kao html dokumenti) i materijala testiranja (html dokumenti sa priloženim JAVA procedurama koje izvode interakciju i vrednuju odgovore). Jedan primjerak nastavnog materijala se može odnositi na više od jednog koncepta odnosno veze i jedan koncept odnosno veza se može prikazati s jednim ili više nastavnih materijala. Za kreiranje nastavnih materijala i materijala testiranja, autor može koristiti bilo koji html uređivač i ponovno upotrijebiti postojeće html izvore. Kreiranje materijala testiranja zahtijeva od autora dodjeljivanje teksta pitanja odnosno problema i točnog odgovora korištenjem predloška JAVA procedure koja izvodi interakciju i vrednuje odgovore.

## UČENJE DCG-om

Učenik šalje zahtjev za kolegijem DCG poslužitelju navođenjem željene domene i cilja poučavanja. DCG učeniku šalje preliminarni test za inicijalizaciju modela učenika i unosi ga kao korisnika. Učenik prima individualni plan kolegija i kopiju JAVA programa zvanog *izvršitelj* koji se pokreće na stranici učenika. Izvršitelj testira učenikovo znanje o konceptima i lokalno ažurira model učenika na stranici učenika. U slučaju da učenik ne završi uspješno test vezan uz određene koncepte koji su potrebni za nastavljanje plana, izvršitelj surađuje sa DCG stranicom - šalje kopiju modela učenika i od planera traži novi plan (prikladan za novo stanje znanja učenika) (Vassileva i Deters, 1998). Proces učenja je prikazan na slici 4.7.



Slika 4.7 Proces učenja DCG-om (modificirano prema (Vassileva i Deters, 1998))

### 4.6.1 Arhitektura DCG-a na WWW-u

Glavna ideja DCG-a je primijeniti tehnike planiranja umjetne inteligencije za kreiranje sadržaja plana kolegija koji ostvaruje određene ciljne koncepte počevši od skupa inicijalnih koncepata. Za ovu namjenu DCG koristi eksplicitni prikaz strukture domene koji je odvojen od nastavnih materijala. Struktura domene je prikazana I-ILI grafom gdje čvorovi (*eng. nodes*) prikazuju koncepte, a veze (*eng. links*) odgovaraju odnosu između koncepata. S danim ciljnim konceptom kojeg učenik želi usvojiti i modelom učenika koji sadrži već poznate koncepte (inicijalizirani pred-testom), AI planer program traži plan (putanju u grafu) koji povezuje učeniku poznate koncepte sa ciljnim konceptom (slika 4.1). Jedan od tih planova se odabire i služi kao kostur kolegija. Za svaki koncept, prikazuju se nastavni materijali (iz baze podataka nastavnih materijala) i kao rezultat učenik vidi poredak nastavnih materijala koji izgleda kao CAL kolegij. Učenika se testira na svakom konceptu plana te se ažurira njegov model učenika. Ako učenik ne ostvari potreban broj bodova za daljnji nastavak plana, odvija se ponovno planiranje kako bi se pronašao alternativni plan za postizanje cilja poučavanja.

Obzirom da je posvojena klijent - poslužitelj arhitektura, neke se komponente DCG-a drže samo na DCG poslužitelju (struktura domene i planer), neke komponente postoje i na DCG poslužitelju i kod klijenta (model učenika i izvršitelj), a neke su raširene diljem Web lokacija (nastavni materijali) (Vassileva i Deters, 1998).

#### STRUKTURA DOMENE

Struktura domene (*eng. domain structure*) se drži na DCG poslužitelju. Prikazana je kao I-ILI graf koji se sastoji od koncepata domene povezanih različitim semantičkim vezama. Jednosmjerne  $1 : n$  i  $n : 1$  veze se koriste za prikaz hijerarhije u odnosu na apstrakciju, agregaciju i posljedične veze. Dvosmjerne  $1 : 1$  veze se koriste za prikaz analogije i privremene veze. Za različite domene, važne su različite semantičke veze, primjerice u tehničkim domenama, vrsta veze koja se najčešće javlja je agregacija. U proceduralnim domenama su važnije uzročne veze i veze preduvjeta. Planiranje se odvija u odnosu na jednu ili više vrsta veza, ovisno o cilju poučavanja. Cilj poučavanja se definira ukazivanjem na ciljni koncept i dodjelom vrste veza u odnosu na plan. Sustav za svaku domenu nudi skup mogućih značajnih ciljeva u meniju iz kojeg učenik mora odabrati. Velika razlika između DCG-a i drugih pristupa za prilagodljivu podršku navigaciji se očituje u većoj izražajnoj moći I-ILI grafova koji omogućavaju prikaz ne samo preduvjeta nego i različitih vrsta relacija među konceptima. Ovo dozvoljava korištenje tehnika planiranja umjetne inteligencije za generiranje kolegija u skladu sa različitim semantičkim vezama. Na ovaj način je osigurana šarolikost različitih ciljeva poučavanja i mogućih kolegija za postizanje tih ciljeva.

#### NASTAVNI MATERIJALI I MATERIJALI TESTIRANJA

Nastavni materijali (*eng. teaching materials*) su html dokumenti koji mogu biti distribuirani diljem Web lokacija. U fazi oblikovanja su dani http linkovi koncepata koji vode do željenih html dokumenata. Primjerci testova materijala testiranja (*eng. testing materials*) se moraju definirati za svaki koncept sa naznakom o njihovoj težini i koeficijentom koji pokazuje u kojoj mjeri (ne)točan odgovor pridonosi cjelokupnom uspjehu koncepta (odnosno veze) u modelu učenika.

## MODEL UČENIKA

Postoje dvije instance modela učenika (*eng. student model*). Jedna instanca je dinamička i postoji na stranici klijenta (učenika). Kopija ove instance se pohranjuje na DCG poslužitelju svaki put kad učenik završi sesiju ili je potrebno ponovno planiranje. Koristi se za pohranjivanje statistike učenikovog uspjeha sa različitim konceptima što je korisno za unapređivanje nastavnih materijala i strukture domene (npr. dekomponiranje složenog koncepta). Ako učenik to želi, model učenika se može anonimno pohraniti na DCG stranicu. Trenutno stanje znanja učenika je prikazano kao pokrivanje strukture domene u bazi podataka, sadrži procjenu sustava da učenik zna određeni koncept. Procjena se gradi korištenjem jednostavne formule s brojem i težinom uspješno riješenih primjeraka testova vezanih uz određeni koncept.

## PLANER

Planer (*eng. planner*) generira plan kolegija na osnovu prikaza strukture domene I-ILI grafom, cilja poučavanja (uzimajući u obzir ciljni koncept i vrste veza) i skupa inicijalnih čvorova (model učenika predstavlja učenikovo trenutno znanje). Algoritam planiranja je modifikacija dobro poznatog algoritma AO\* (Nilsson, 1980). Plan je prikazan kao pod-graf strukture domene koji se sastoji od konceptata i veza koje se moraju tijekom kolegija poučavati. Izgrađen je linearno s lijeva na desno, rezultat je lista konceptata za poučavanje. Pedagoški opravdaniji način postizanja linearnosti je korištenje pravila poučavanja temeljenih na dekomponiranju zadatka koji će se u bliskoj budućnosti implementirati u DCG na WWW-u.

## IZVRŠITELJ

Izvršitelj plana (*eng. plan executor*) je učitana na učenikovu stranicu na početku sesije. Za ulaz uzima plan i generira kolegij traženjem prikladnih nastavnih materijala na WWW-u (html dokumenti pridruženi konceptima domene), učitavanjem istih na učenikovu stranicu i njihovom prezentiranjem učeniku. Kada učenik želi pristupiti testiranju znanja o konceptu, izvršitelj odabire primjerak testa, prikaže ga učeniku te prihvaća i vrednuje učenikove odgovore i primjereno tome ažurira model učenika. Ako razina znanja o konceptu koja je nužna za uspjeh plana nije dovoljna, može ponovno pozvati planera za kreiranje novog plana. U ovom slučaju, sadržaj modela učenika se šalje natrag na DCG stranicu kako bi se uzeo u obzir kod ponovnog planiranja.

## 5. Model višeagentskog sustava za dinamičko generiranje courseware-a

Glavna zamisao DCG-a je dinamičko planiranje nastavnih sadržaja, pri čemu isporuka tih nastavnih sadržaja ovisi o samom učeniku i njegovom razumijevanju. Za optimiziranje procesa učenja i poučavanja potrebno je uvažavanje individualnih razlika učenika, tj. prilagodljivo ponašanje sustava s ciljem rješavanja problema, stoga se uvode inteligentni agenti koji vode računa o odabiru i o određivanju redoslijeda nastavnih materijala. Zašto inteligentni agenti? Za ovaj specifični problem, agentima je potrebno svojstvo inteligencije koje ih opskrbljuje uspješnim snalaženjem u novim situacijama. Inteligentni agenti imaju svojstva učenja, personalizacije i prilagodljivosti čime osiguravaju pedagošku primjerenost načina prikaza nastavnog sadržaja za određenog učenika. Kako inteligentni agenti imaju svojstvo suradnje, za jednostavnije korištenje sustava uvodimo višeagentski sustav u kojem inteligentni agenti surađuju u svrhu postizanja cilja.

U sustavima e-učenja fiksna putanja (*eng. fixed pathway*) nije prikladna za sve učenike pa se pažnja treba usmjeriti na određivanje redoslijeda kolegija (*eng. course sequencing*). Iako većina sustava e-učenja pri određivanju redoslijeda kolegija uzima u obzir učenikove preferencije, interese i ponašanje, ne uzimaju u obzir usklađenost učenikovog znanja i težine nastavnog sadržaja. Neprikladni nastavni sadržaj vodi do kognitivnog preopterećenja ili do dezorijentiranosti tijekom učenja, čime je smanjen učinak učenja (Chen i ostali, 2006).

Ideja određivanja redoslijeda kolegija je generiranje individualiziranog kolegija za svakog učenika dinamičkim odabirom optimalne operacije poučavanja (*eng. optimal teaching operation*) koja uvažava različito predznanje, preferencije i često različite ciljeve poučavanja. Optimalnom operacijom poučavanja se smatra ona operacija koja u sklopu ostalih dostupnih operacija dovodi učenika najbliže osnovnom (*eng. ultimate*) cilju poučavanja (Brusilovsky i Vassileva, 2003).

### 5.1 Arhitektura sustava

Zbog kvalitete učenja i napretka individualnog učenika, courseware se odabire tako da pristaje individualnim potrebama učenika. Takav odabir ovisi o samom učeniku i njegovim prethodnim interakcijama sa courseware-om, tj. o informacijama o učeniku sadržanima u modelu učenika. Neodgovarajući courseware će samo usporiti učenje i napredak, stoga sustav uzima u obzir model učenika kao i težinske razine courseware-a. Provjerava prikladnost težinske razine courseware-a učenikovom znanju. Dinamički se procjenjuje znanje učenika prikupljanjem učenikove povratne informacije nakon učenja odabranog courseware-a. Na osnovu procjene znanja učenika, sustav nudi courseware s prikladnom težinskom razinom.

Predložena je arhitektura sustava koja je modificirana prema (Chen i ostali, 2006), a se sastoji od sljedećih komponenti (slika 5.1):

- 1) Baza podataka korisnika
- 2) Baza podataka modela učenika
- 3) Baza podataka courseware-a
- 4) Agent sučelja

Namjena agenta sučelja je pružiti učenicima fleksibilno sučelje za učenje i za interakciju s agentom modeliranja i pedagoškim agentom.

- 5) Agent modeliranja

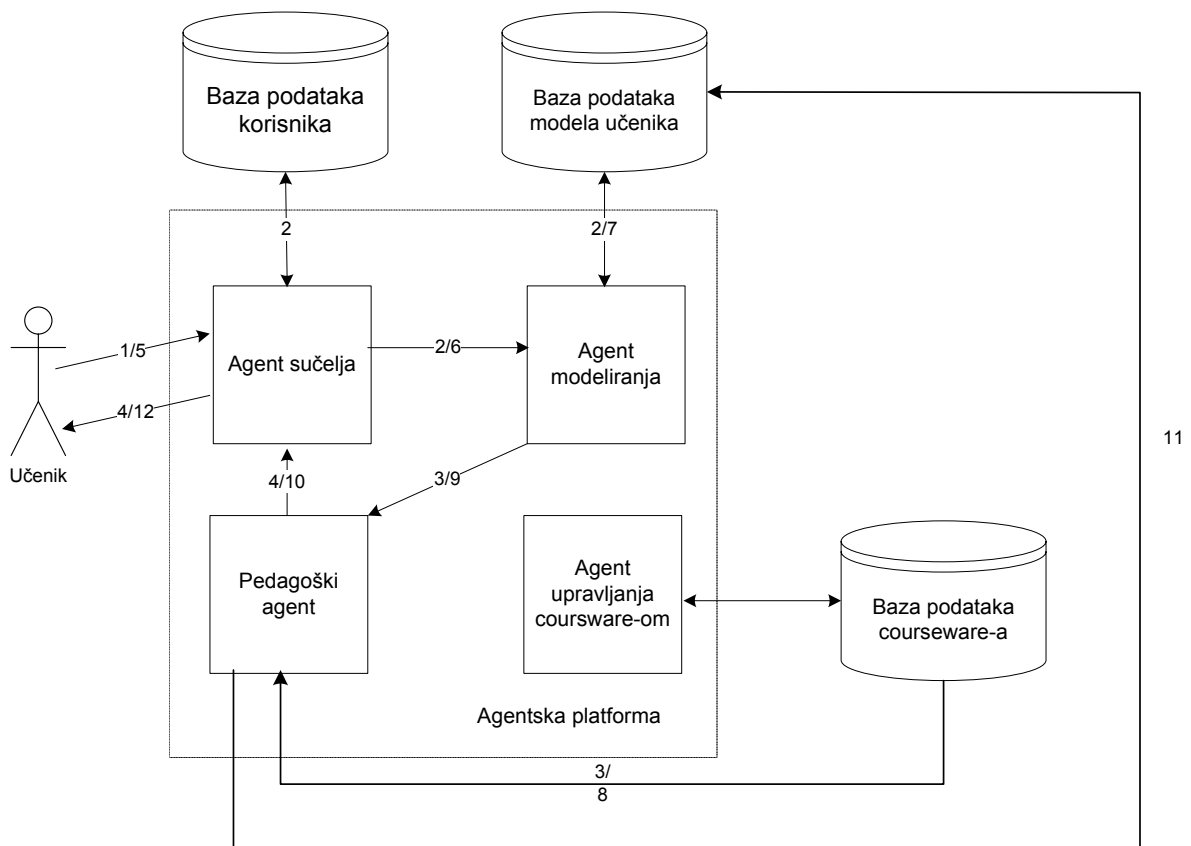
Agent modeliranja je odgovoran za održavanje, ažuriranje i analiziranje modela učenika koji se koristi za generiranje courseware-a prilagođenog individualnom učeniku.

- 6) Pedagoški agent

Pedagoški agent je zadužen za odabir plana kolegija na osnovu učenikove povratne informacije i težinske razine courseware-a.

- 7) Agent upravljanja courseware-om

Agent upravljanja courseware-om je odgovoran za održavanje, ažuriranje i administriranje baze podataka courseware-a.



Slika 5.1 Prijedlog arhitekture sustava temeljen na inteligentnim agentima za dinamičko generiranje courseware-a (modificirano prema (Chen i ostali, 2006))

Stručnjaci oblikuju courseware i jedinice testiranja za sadržaj učenja. Težinske razine tih jedinica se mogu odrediti pomoću statističkih metoda. Dizajnirani courseware se održava pomoću agenta upravljanja courseware-om i pohranjuje se u bazu podataka courseware-a.

Učitelji pristupaju sustavu za dodavanje, brisanje ili mijenjanje courseware-a u bazu podataka courseware-a pomoću svog korisničkog računa. Učitelj održava bazu podataka courseware-a pomoću agenta upravljanja courseware-om.

Na osnovu predložene arhitekture sustava (slika 5.1), opisan je rad sustava sljedećim koracima:

1. Učenik se prijavljuje na sustav pomoću agenta sučelja.
2. Pri prijavljivanju učenika na sustav, agent sučelja provjerava njegov korisnički račun u bazi podataka korisnika. Ako je učenik već registriran, šalje njegove podatke agentu modeliranja koji će dohvatiti njegov model učenika iz baze podataka modela učenika za pružanje usluge personaliziranog učenja, inače će izgraditi model učenika za ovog učenika i pohraniti ga u bazu podataka modela učenika.
3. Kako pedagoški agent prima informacije o courseware-u iz baze podataka courseware-a i o učeniku iz baze podataka modela učenika (od agenta modeliranja), tako odabire odgovarajući courseware i šalje listu odabranih courseware-a agentu sučelja.
4. Agent sučelja prima predloženi courseware iz baze podataka courseware-a i izlaže ga učeniku.
5. Učenik mora odgovoriti na upitnik vezan uz težinsku razinu i postotak shvaćanja učenog courseware-a.
6. Agent sučelja prosljeđuje učenikovu povratnu informaciju agentu modeliranja.
7. Agent modeliranja sprema učenikove povratne odgovore u bazu podataka modela učenika, tj. ažurira model učenika.
8. Pedagoški agent ima uvid u težinske parametre učenog courseware-a iz baze podataka courseware-a.
9. Pedagoški agent vrednuje učenikovo znanje u skladu sa povratnim odgovorima učenika koje mu je proslijedio agent modeliranja.
10. Pedagoški agent računa odgovarajuću funkcijsku vrijednost informacije za svaki courseware o učenim jedinicama courseware-a u skladu sa vrednovanim učenikovim znanjem i težinskim razinama courseware-a. Tada rangira sve courseware-e po redoslijedu funkcijskih vrijednosti informacije. Predaje rezultate agentu sučelja, tj. pruža listu preporučenih courseware-a za učenika.
11. Informacija o učeniku (koja uključuje znanje, učenikov odgovor i povijest), izračunata od strane pedagoškog agenta se pohranjuje u bazu podataka modela učenika.
12. Agent sučelja pokazuje listu preporučenog courseware-a za učenika i čeka na njegov povratni odgovor. Nakon što učenik odabere sljedeći courseware za daljnje učenje, operativna procedura sustava se vraća na 3. korak, ponavlja korake od 3. do 12. sve dok se učenik ne odjavi sa sustava.

U ovom prijedlogu arhitekture sustava sa četiri inteligentna agenta ne dolazi do interakcije svih agenata, primjerice agent upravljanja courseware-om nije u interakciji ni s jednim od preostala tri agenta. Iako za rad višeagentskog sustava nije nužna interakcija svih agenata, bilo bi poželjno dovesti i agenta upravljanja courseware-om u interakciju s preostalim agentima što će se detaljnije obraditi u točki 5.2.5.

Kako bismo dali različite poglede na arhitekturu sustava (pogled korištenja sustava i logički pogled oblikovanja sustava), dat ćemo pregled tri UML<sup>3</sup> dijagrama: dijagram slučajeva korištenja, dijagram klasa i dijagram suradnje. UML dijagram je grafički prikaz skupa odgovarajućih elemenata za vizualizaciju sustava iz različitih perspektiva.

**Dijagram slučajeva korištenja** (eng. *use case diagram*) prikazuje interakciju sudionika sustava sa slučajevima korištenja koji opisuju funkcionalnost sustava, tj. *što* sustav radi.

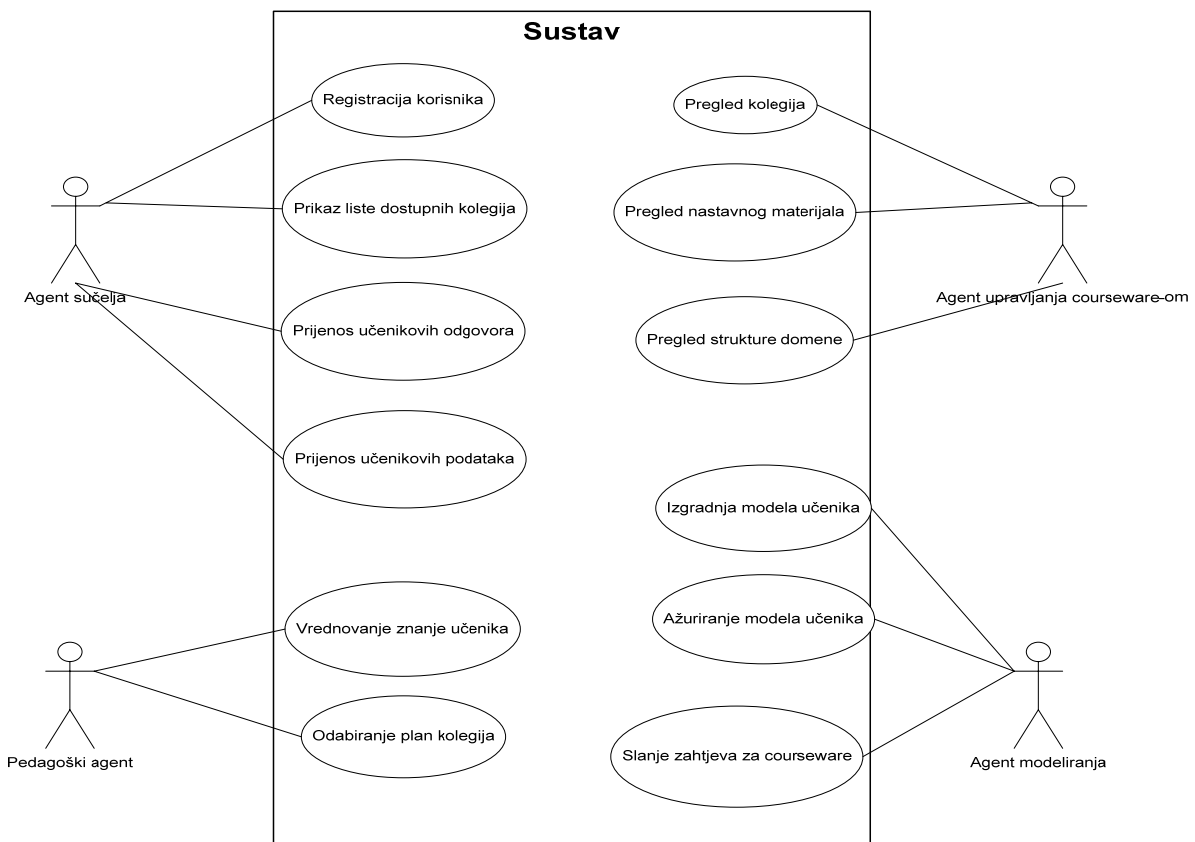
Elementi dijagrama slučajeva korištenja su slučajevi korištenja (eng. *use case*), sudionici (eng. *actors*) te različite veze među njima.

Sudionici ovog sustava su učenik, učitelj, stručnjak i administrator koji su sa slučajevima korištenja povezani vezom pridruživanja (eng. *association*) (slika 5.3). Obzirom da se ovaj sustav sastoji i od četiri inteligentna agenta, to su i agenti sudionici ovog sustava jer sudionik ne mora nužno biti osoba, već i dio sustava (slika 5.2).

Za odnos između dva slučaja korištenja ovog sustava se koriste veze proširivanja (eng. *extend*) i veze uključivanja (eng. *include*).

Primjerice, veza uključivanja od slučaja korištenja *oblikuj čvor* prema slučaju korištenja *preimenuj čvor* ukazuje da će jedan dio slučaja korištenja *oblikuj čvor* uključivati slučaj korištenja *preimenuj čvor*.

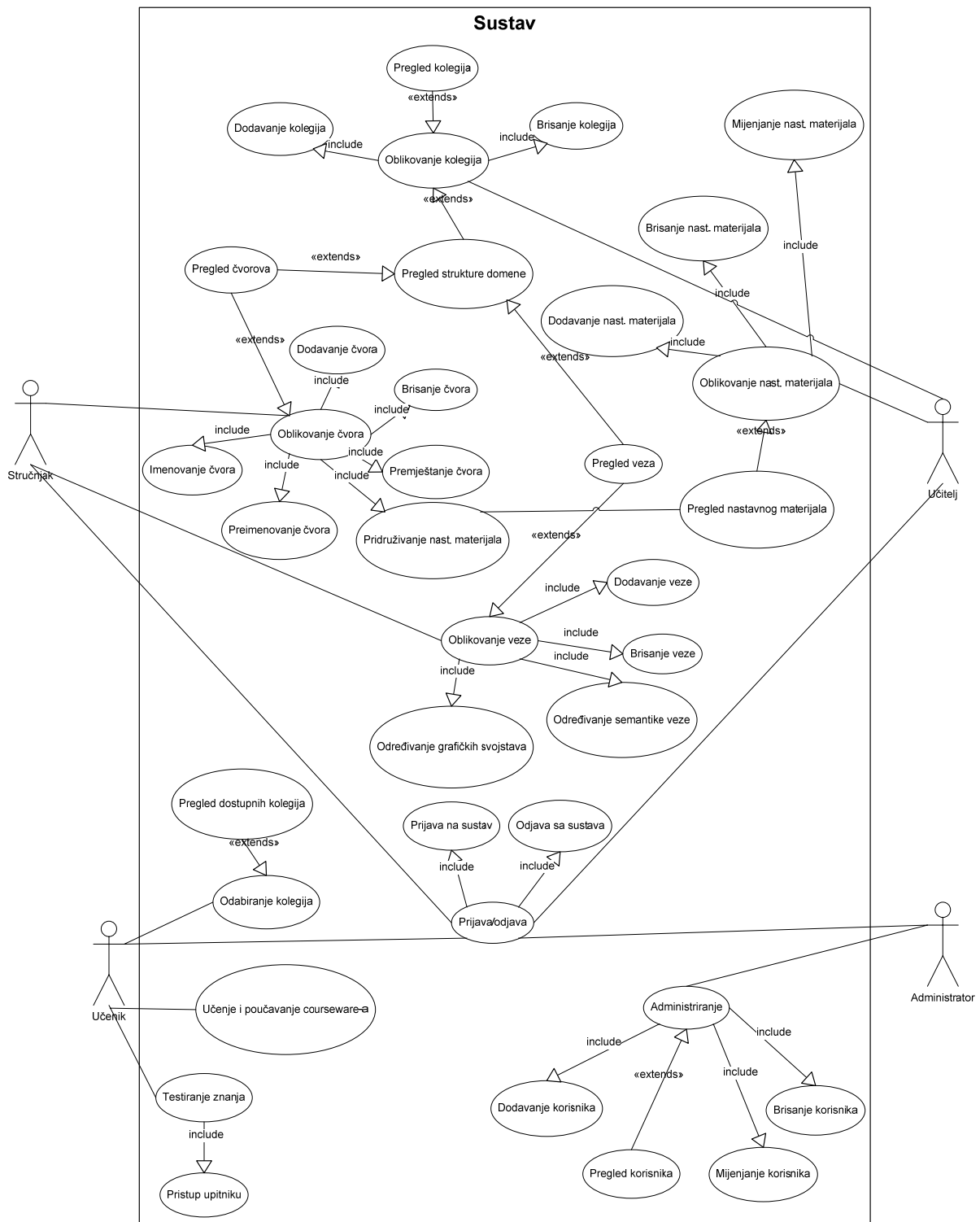
Primjerice, veza proširivanja od slučaja korištenja *pregled čvorova* prema slučaju korištenja *oblikuj čvor* indicira primjerak slučaja korištenja *pregled čvorova* koji je prošireno ponašanje slučaja korištenja specificiran sa *oblikuj čvor*.



Slika 5.2 Dijagram slučajeva korištenja (I. dio)

<sup>3</sup>UML (eng. *Unified Modeling Language*) je grafički jezik za specifikaciju, vizualizaciju, konstruiranje i dokumentiranje komponenti programskog sustava koji je postavljen kao standard od OMG-a (eng. *Object Management Group*).





Slika 5.3 Dijagram slučajeva korištenja (II. dio)

Slučajevi korištenja sudionika *agent sučelja* su registracija korisnika, prikaz liste dostupnih kolegija, prijenos učenikovih odgovora i prijenos učenikovih podataka.

Slučajevi korištenja sudionika *agent modeliranja* su izgradnja modela učenika, ažuriranje modela učenika i slanje zahtjeva za courseware.

Slučajevi korištenja sudionika *pedagoški agent* su vrednovanje znanja učenika i odabiranje plana kolegija.

Slučajevi korištenja sudionika *agent upravljanja courseware-om* su pregled kolegija, pregled nastavnog materijala i pregled strukture domene.

Slučajevi korištenja sudionika *učenik* su prijava/odjava, odabiranje kolegija, učenje i poučavanje courseware-a i testiranje znanja. Slučaj korištenja prijava/odjava uključuje prijavu i odjavu sa sustava, slučaj korištenja testiranje znanja uključuje slučaj korištenja pristup upitniku. Proširenje slučaja korištenja odabiranje kolegija je slučaj korištenja pregled dostupnih kolegija jer će slučaj korištenja odabir kolegija u svom radu preći na pregled dostupnih kolegija.

Slučajevi korištenja sudionika *stručnjak* su prijava/odjava, oblikovanje čvora i oblikovanje veze. Slučaj korištenja oblikovanje čvora uključuje sljedeće slučajeve korištenja koji su pod-zadaci tog slučaja korištenja: dodavanje čvora, brisanje čvora, premještanje čvora, imenovanje čvora, preimenovanje čvora te pridruživanje nastavnog materijala. Proširenje slučaja korištenja oblikovanje čvora je slučaj korištenja pregleda čvorova jer će slučaj korištenja oblikovanje čvorova u svom radu preći na pregled čvorova. Slučaj korištenja oblikovanje veze uključuje dodavanje veze, brisanje veze, određivanje semantike veze, određivanje grafičkih svojstava i njegovo proširenje je slučaj korištenja pregled veza jer će slučaj korištenja oblikovanje veza u svom radu preći na pregled veza. Spomenuti slučajevi korištenja pregled čvorova i pregled veza su proširenja slučaja korištenja pregled strukture domene.

Slučajevi korištenja sudionika *učitelj* su prijava/odjava, oblikovanje nastavnih materijala te oblikovanje kolegija. Proširenje slučaja korištenja oblikovanje nastavnih materijala je slučaj korištenja pregled nastavnog materijala. Slučaj korištenja oblikovanje nastavnih materijala uključuje dodavanje nastavnog materijala, mijenjanje nastavnog materijala i brisanje nastavnog materijala. Slučaj korištenja oblikovanje kolegija uključuje dodavanje i brisanje kolegija, ima dva proširenja: slučaj korištenja pregled strukture domene i slučaj korištenja pregled kolegija.

Slučajevi korištenja sudionika *administrator* su prijava/odjava sa sustava kao i kod sudionika učenik te administriranje koje uključuje sljedeće slučajeve korištenja koji su pod-zadaci tog slučaja korištenja: dodavanje korisnika, mijenjanje korisnika, brisanje korisnika. Proširenje slučaja korištenja administriranje je slučaj korištenja pregled korisnika jer će slučaj korištenja administriranje u svom radu preći na slučaj korištenja pregled korisnika.

**Dijagram klasa** (*eng. class diagram*) opisuje statičku strukturu sustava. Daje pregled sustava pokazujući njegove klase i odnose među tim klasama prilikom logičkog oblikovanja sustava.

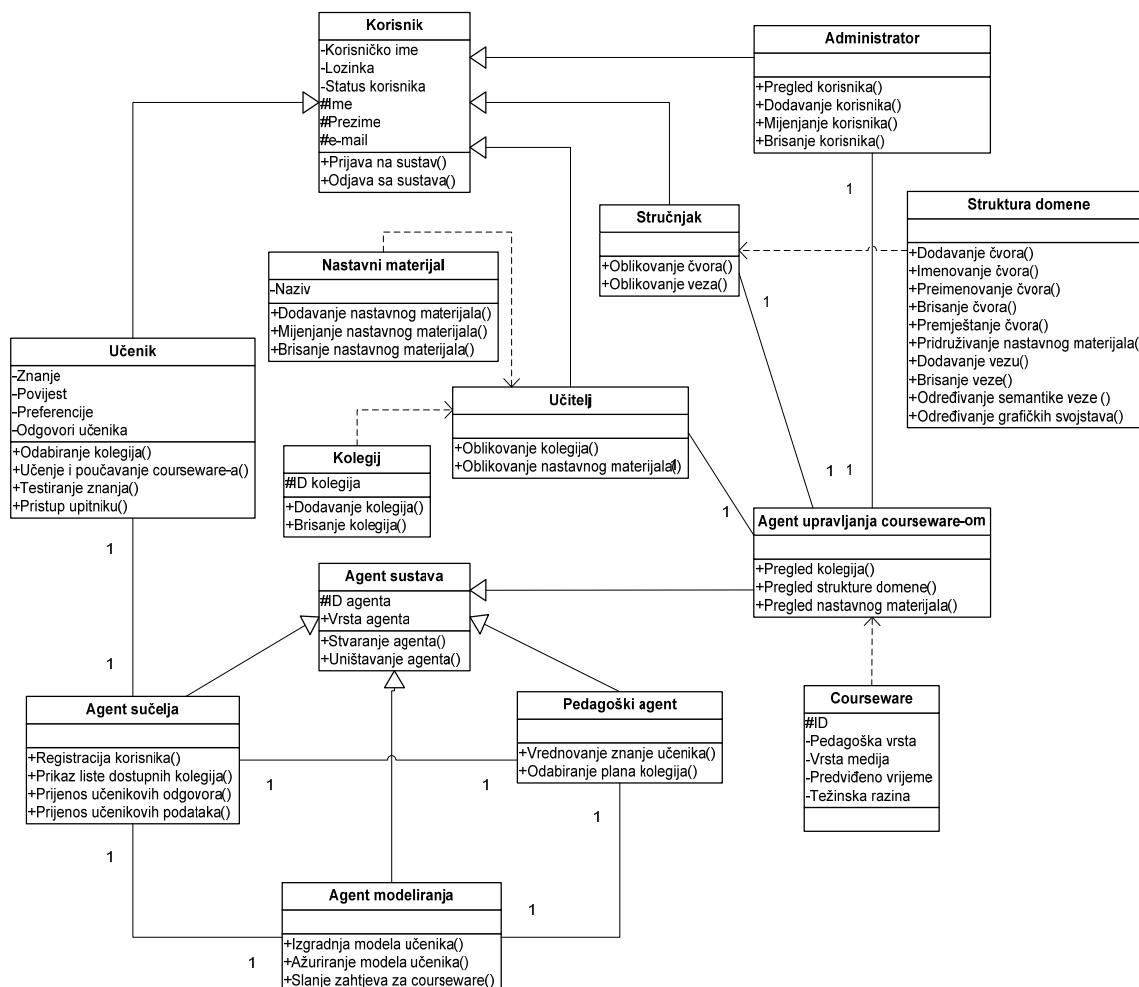
Na ovom dijagramu su označene klase koje su vezane za sustav, njihovi atributi i metode (slika 5.4). Primjerice klasa *Agent sustava* ima attribute *ID agenta* i *vrsta agenta* te metode *stvari agenta* i *uništi agenta*.

Za opis odnosa između klasa su korištene veze generalizacije (*eng. generalization*) i pridruživanja (*eng. association*) te zavisnosti (*eng. dependency*).

Generalizacija opisuje hijerarhijski odnos među klasama, primjerice, klasa *Korisnik* je generalizacija klasa *Učenik*, *Učitelj*, *Stručnjak* i *Administrator*.

Veza pridruživanja kaže da jedan objekt ima za atribut primjerak drugog ili su ti objekti povezani u smislu posjedovanja (ali ne i odnosa *sastojati se od*), primjerice učenik je vezan pridruživanjem sa agentom sučelja, ali se učenik ne sastoji od agenta sučelja.

Zavisnost prikazuje kako promjena na jednoj od klasa utječe na drugu klasu, primjerice klasa *Kolegij* zavisi o klasi *Učitelj*.



Slika 5.4 Dijagram klasa

Klasa *Korisnik* ima podklase *Učenik*, *Učitelj*, *Stručnjak* i *Administrator* koje nasljeđuju atribute i metode svoje nadklase korisnik.

Klasa *Agent sustava* ima podklase *Agent sučelja*, *Agent modeliranja*, *Pedagoški agent* i *Agent upravljanja courseware-om* koje nasljeđuju atribute svoje nadklase agent sustava. Klasa *Agent sustava* realizacijom svojih metoda stvaranje agenta i uništavanje agenta upravlja životnim ciklusom svojih podklasa.

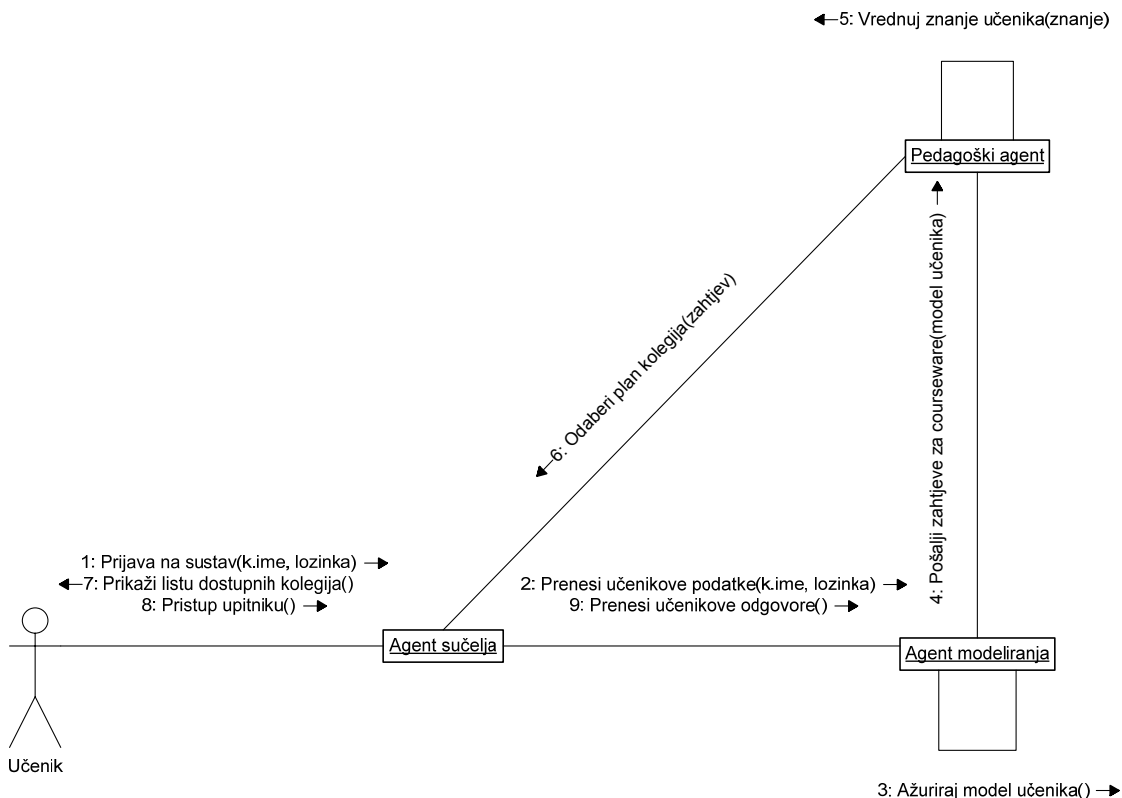
Klase *Učenik* i *Agent sučelja* su vezane pridruživanjem koja prikazuje suradnju tih dviju klasa (mnogostrukost pridruživanja je 1 prema 1, tj. na jednog učenika ide jedan agent sučelja). Pridruživanjem (mnogostrukosti 1 prema 1) su vezani i sljedeći parovi klasa: *Agent sučelja* i *Agent modeliranja*, *Agent modeliranja* i *Pedagoški agent*, *Pedagoški agent* i *Agent sučelja*, *Učitelj* i *Agent upravljanja courseware-om*, *Stručnjak* i *Agent upravljanja courseware-om*, *Administrator* i *Agent upravljanja courseware-om*.

Promjena klase *Učitelj* (pri čemu u obzir treba uzeti da su klase *Učitelj* i *Agent upravljanja courseware-om* vezane pridruživanjem) utječe na klasu *Nastavni materijal* i klasu *Kolegij*, promjena klase *Stručnjak* (pri čemu u obzir treba uzeti da su klase *Stručnjak* i *Agent upravljanja courseware-om* vezane pridruživanjem) utječe na klasu *Struktura domene*, promjena klase *Agent upravljanja courseware-om* utječe na klasu *Courseware* jer su te klase povezane vezom zavisnosti.

**Dijagram suradnje** (eng. *collaboration diagram*) je vrsta dijagrama interakcije koji naglašava odnose između objekata i protok poruka između objekata. Pokazuje objekte koji sudjeluju u interakciji i slijed razmijenjenih poruka. Obzirom da dijagram suradnje ne pokazuje vrijeme kao zasebnu dimenziju, redosljed poruka određuju brojevi slijeda na porukama.

Dijagram suradnje predstavlja suradnju koja sadrži skup uloga koje igraju instance kao i potrebne veze dane u određenom kontekstu. Suradnja se koristi da bi se prikazala realizacija metode. Ona je skup sudionika i veza koje imaju smisla unutar konteksta.

Dijagram suradnje opisuje specifični scenarij, u ovom slučaju prikazuje proces učenja i poučavanja koji se ostvaruje realizacijom metoda klasa *Učenik*, *Agent sučelja*, *Agent modeliranja* i *Pedagoški agent* (slika 5.5).



Slika 5.5 Dijagram suradnje koji prikazuje proces učenja i poučavanja

Suradnja učenika i agenata u procesu učenja i poučavanja teče tako da nakon što se učenik prijavi na sustav, Agent sučelja prenosi njegove podatke (korisničko ime i lozinku) Agentu modeliranja. Agent modeliranja ažurira model učenika i šalje zahtjev za courseware Pedagoškom agentu. Pedagoški agent vrednuje znanje učenika i uzima u obzir težinsku razinu courseware-a te odabire plan kolegija. Pedagoški agent šalje plan kolegija Agentu sučelja koji učeniku prikazuje listu dostupnih kolegija.

## 5.2 Inteligentni agenti u dinamičkom generiranju courseware-a

Agenti surađuju s ciljem ostvarivanja svog zadatka, tj. odabiranja courseware-a odgovarajuće težinske razine koji će se isporučiti učeniku te vrednovanja učenikovog uspjeha s odabranim courseware-om.

### 5.2.1 Agent sučelja

Agent sučelja detektira interakcije učenika sa sučeljem sustava te pohranjuje i/ili prosljeđuje rezultate tih interakcija. Pruža prijateljsko sučelje za učenje, prenosi učenikove povratne informacije agentu modeliranja, prima rezultate biranja courseware-a od pedagoškog agenta. Zahvaljujući agentu sučelja, učenici mogu izabrati zanimljive kategorije kolegija.

Ukoliko učenik po prvi put pristupa sustavu, mora se registrirati (ime, prezime, e-mail adresa). Kada se učenik prvi put prijavi na sustav, agent sučelja izdvaja dano korisničko ime i lozinku i prosljeđuje te podatke agentu modeliranja. Nakon učenikove prijave na sustav, agent sučelja odabire courseware s umjerenom težinom i učenik mora odgovoriti na dva upitnika. Kada učenik ispuni upitnik, agent sučelja predaje odgovore agentu modeliranja. Ta povratna informacija se koristi za izgradnju modela učenika.

Ukoliko učenik već ima iskustva u radu sa sustavom, sustav na osnovu modela učenika preporučuje prikladni courseware.

### 5.2.2 Agent modeliranja

Kako bi što više pojednostavnili mehanizme za prilagođavanje courseware-a učeniku, učenik mora odgovoriti na upitnik vezan uz težinsku razinu i postotak shvaćanja učenog courseware-a (ako učenik razumije sadržaj učenog courseware-a, njegovo znanje će se rasti, obrnuto, ako učenik NE razumije sadržaj učenog courseware-a, njegovo znanje će opadati).

Namjena agenta modeliranja je prikupiti te eksplicitne povratne informacije od agenta sučelja i pohraniti ih u bazu podataka modela učenika. Koristi skup pravila za ocjenjivanje potreba učenika kako bi te zahtjeve mogao proslijediti pedagoškom agentu za preporučivanje odgovarajućeg courseware-a učeniku. Ta pravila usmjeruju ponašanje cijelog sustava i osiguravaju personalizaciju koja neće ometati učenika u procesu stjecanja znanja.

### 5.2.3 Pedagoški agent

Kada agent modeliranja detektira određeni zahtjev/potrebu, tu informaciju šalje pedagoškom agentu koji odlučuje o ponovnom planiranju. Dakle, pedagoški agent je zadužen za odabir plana kolegija na osnovu učenikove povratne informacije i težinske razine courseware-a. Pedagoški agent prvo procjenjuje znanje učenika pomoću neke procedure procjenjivanja, zatim rangira courseware u skladu sa težinskom razinom za preporučivanje courseware-a te ovisno o podudaranju znanja učenika i težinske razine courseware-a, šalje odabir courseware-a agentu sučelja koji taj odabir prikazuje učeniku.

### 5.2.4 Agent upravljanja courseware-om

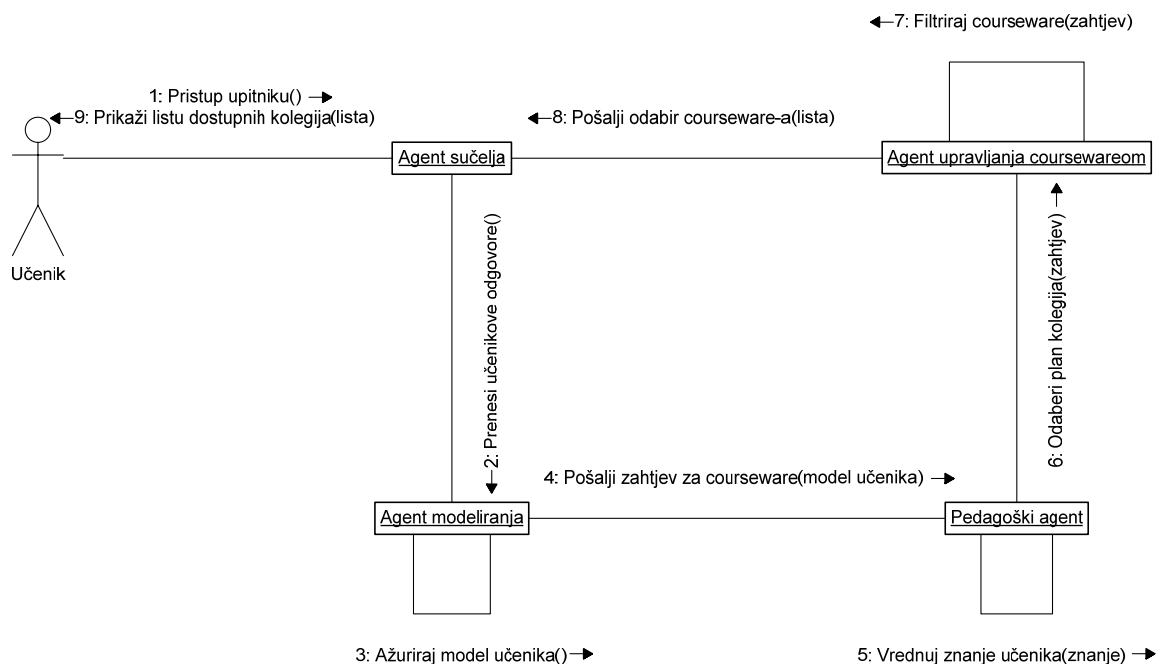
Agent upravljanja courseware-om pruža sučelje za učitelje, stručnjake i administratore sustava. Agent upravljanja courseware-om upravlja detaljima održavanja baze podataka courseware-a. Omogućava sučelje za razvijanje courseware-a, pregled razvijenog courseware-a, dodavanje, uređivanje i brisanje strukture i sadržaja courseware-a u bazi podataka courseware-a. Administratori sustava upravljaju računima stručnjaka, učitelja i učenika.

### 5.2.5 Suradnja agenata

Kako bi se učenik koristio sustavom komunicira sa Agentom sučelja koji mu pruža odgovarajuće sučelje za odabir kolegija i učenja odabranog courseware-a te pristup upitniku. Nakon odabira kolegija i učenja odgovarajućeg courseware-a, učenik pristupa upitniku vezanom za učeni courseware. Agent sučelja prihvaća učenikove odgovore i šalje ih Agentu modeliranja koji ih pohranjuje u model učenika. Nakon analiziranja i ažuriranja modela učenika, Agent modeliranja šalje zahtjeve za učenika sadržane u modelu učenika Pedagoškom agentu koji na osnovu svojih internih mehanizama vrednuje znanje učenika i donosi odluku o tome koja je težinska razina odgovarajuća za učenika. Pedagoški agent šalje svoju odluku Agentu upravljanja courseware-om koji odluku Pedagoškog agenta koristi kao kriterij za filtriranje baze podataka courseware-a i rezultat šalje Agentu sučelja koji taj rezultat prikazuje učeniku.

Dakle, imamo sljedeće interakcije agenata (slika 5.6):

- Agent sučelja – Agent modeliranja
- Agent modeliranja – Pedagoški agent
- Pedagoški agent – Agent upravljanja courseware-om
- Agent upravljanja courseware-om – Agent sučelja



Slika 5.6 Dijagram suradnje agenata u dinamičkom generiranju courseware-a

Primjerice, učenik se prijavljuje na sustav i pomoću Agentu sučelja odabire kolegij „Vizualno modeliranje“. Agent modeliranja šalje zahtjev za personalizirano učenje za tog učenika Pedagoškom agentu koji u skladu s znanjem učenika (pohranjenim u modelu učenika) na osnovu svojih internih mehanizama donosi odluku o tome što će se tog učenika poučavati. Potom Pedagoški agent šalje svoje zahtjeve Agentu upravljanja courseware-om koji šalje odabrani courseware Agentu sučelja koji će taj courseware prikazati učeniku na zaslonu njegovog računala. Recimo da je Pedagoški agent zatražio courseware o UML dijagramima.

Učenik uči o UML dijagramima te nakon učenja tog odabranog courseware-a, učenik pristupa upitniku vezanom za učenici courseware-a kolegija. Ukoliko je učenik zadovoljio prag dovoljnosti, tj. smatra se da je usvojio to gradivo, opet dolazi do suradnje agenata koji na prethodno opisani način opet odabiru courseware za učenika iste ili veće težinske razine o sljedećem konceptu tog kolegija, recimo za UML dijagram slučajeve korištenja.

Ukoliko nakon učenja tog odabranog courseware-a o UML dijagramu slučajeve korištenja i pristupa upitniku vezanom za učenici courseware-a kolegija učenik ne zadovolji, agenti opet surađuju u odabiranju courseware-a za taj isti koncept (UML dijagram slučajeve korištenja), pri čemu se na različitim prikazom istog znanja nastoji poučavati učenika (primjerice korištenje nastavnih materijala različite pedagoške ili medijske vrste te drukčiji tijek istih).

## 6. Zaključak

Jedan od najvažnijih ciljeva obrazovanja jest prenijeti znanje na učenika. Tradicionalni način poučavanja nije usmjeren na individualnog učenika, već na razred koji se sastoji od nekih trideset učenika. Znamo da učenici na različite načine usvajaju i obrađuju informacije i to različitom tempom, tj. svaki učenik ima odgovarajući stil učenja, stoga ne čudi što proces poučavanja ne vodi uvijek do učenja. Od velike je važnosti da učitelji optimiziraju svoj način poučavanja, no teško je odabrati i primijeniti odgovarajući način poučavanja za trideset učenika.

Razvojem WWW-a je došlo do promjena i u obrazovanju. WWW je izvor informacija koje su nam dostupnije nego ikad. Izgrađeno je više Web-utemeljenih sustava obrazovanja, no zanimaju nas oni sustavi koji uvažavaju individualnost onog koji uči, tj. prilagodljivi sustavi koji dinamički generiraju courseware kako bi što bolje odgovarao učenikovim potrebama. Novi način poučavanja je usmjeren na individualnog učenika. DCG je procijenjen kao samostalni autorski alat u nekoliko domena i pokazane su njegove prednosti i primjene. ITS autorski alat sa distribuiranom arhitekturom poput DCG-a može pružiti dobar početak prema Web-utemeljenim prilagodljivim sustavima obrazovanja.

Ovaj rad predstavlja razvijanje arhitekture za dinamičko generiranje courseware-a – DCG koja omogućava eksplicitni prikaz stručnog poučavanja. Nastavni kolegij se automatski generira za dani cilj i može se dinamički mijenjati u skladu sa određenim pravilima poučavanja kako bi što bolje pristajao učenikovom individualnom napretku i potrebama. Arhitektura sustava se temelji na eksplicitnom prikazu strukture domene, tj. onog što mora biti usvojeno i obrazovnih zadataka i metoda, tj. način poučavanja. Zasebni prikaz nastavnih materijala omogućava veću fleksibilnost i individualizaciju, kao i jednostavnije ažuriranje i ponovnu upotrebu gotovih materijala.

U ovom radu je opisana primjena višeagentskog sustava u području obrazovanja. Posebna pažnja je posvećena DCG-u koji je korišten prilikom implementacije višeagentskog sustava za učenje i poučavanje kako bi se uspješno udovoljilo zahtjevu za prilagodljivost učeniku. Primjena višeagentskih sustava koji se sastoje od inteligentnih agenata, pokazala se pogodnom za automatizaciju procesa učenja i poučavanja. Mogućnost suradnje omogućuje agentu da po potrebi komunicira s drugim agentima s ciljem realizacije zajedničkoga cilja. Zbog svojih osobina, višeagentski sustavi s inteligentnim agentima predstavljaju učinkovito rješenje za ostvarivanje prilagodljivosti učeniku u procesu učenja i poučavanja.



## 7. Literatura

- Brusilovsky, P. i Vassileva, J. (2003) „Course sequencing techniques for large-scale web-based education“, Int. J. Continuing Engineering Education and Lifelong Learning, Vol. 13, Nos. 1/2, pp.75-94.
- Chen, C., Liu, C. i Chang M. (2006) „Personalized curriculum sequencing utilizing modified item response theory for web-based instruction“, Expert Systems with Applications 30 (2006) 378-396
- Mispelkamp, H. i Sarti, L. (1995) DISCourse: Tools for the design of learning material. Materials Science and Engineering, 1 August 1995, vol. 199 no. 1, pp. 99-107(9).
- Nilsson, N. (1980) „Principles of Artificial Intelligence“, San Francisco: Morgan Kaufmann
- Rosić, M. (2000) „Zasnivanje sustava obrazovanja na daljinu unutar informacijske infrastrukture“, magistarski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, Zagreb
- Stankov, S. (2007) Nastavni materijali iz kolegija Primjena računala u nastavi, Prirodoslovno-matematički fakultet, Sveučilište u Splitu, [URL:http://proliant.pmfst.hr/stankov/Nastavni\\_Materijal\\_PDSFPMZ](http://proliant.pmfst.hr/stankov/Nastavni_Materijal_PDSFPMZ) (17/01/2009)
- Šuljug, A. (2008) „Inteligentni agenti za pretraživanje Web-a“, diplomski rad, Prirodoslovno – matematički fakultet, Sveučilište u Splitu
- Vassileva, J. (1994) „A New Approach to Authoring of Adaptive Courseware for Engineering Domains“, Institut für Technische Informatik, Universität der Bundeswehr München, Germany
- Vassileva, J. (1995) „Dynamic Courseware Generation: at the Cross Point of CAL, ITS and Authoring“, Institut für Technische Informatik, Universität der Bundeswehr München, Germany
- Vassileva, J. (1997) „Dynamic Courseware Generation“, Institut für Technische Informatik, Universität der Bundeswehr München, Germany
- Vassileva, J. (1998) „Dynamic Courseware Generation with Teaching Expertise“, Institut für Technische Informatik, Universität der Bundeswehr München, Germany
- Vassileva, J. i Deters R. (1998) „Dynamic Courseware Generation on the WWW“, Institut für Technische Informatik, Universität der Bundeswehr München, Germany
- Xxx: UML. [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm#UML](http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML) (14/02/2009)