

# Razvoj sistema za proveru znanja studenata u okvirima postojećeg IS univerziteta

Goran P. Šimić<sup>1</sup>, Aleksandar D. Jevremović<sup>2</sup>, Vojna Akademija Beograd, Univerzitet Singidunum Beograd

**Sadržaj** — Ovaj rad ima za cilj da predstavi razvoj UNIS.MTutor sistema za proveru znanja studenata. Postoji nekoliko grupa sistema za elektronsko učenje. Njihovi nedostaci su: domenska specijalizovanost, teško prilagođavanje gotovih rešenja konkretnim potrebama obrazovnih institucija, i ograničene mogućnosti proveravanja. Ove karakteristike kao i prethodna istraživanja, rezultati i iskustvo u razvoju predstavljalo je osnovne motive za razvoj vlastitog sistema za proveru znanja u okviru postojećeg informacionog sistema univerziteta. U radu su prikazani najznačajni detalji dizajna sistema i evaluirana iskustva iz praktičnog korišćenja sistema.

**Ključne reči** — provera znanja, sistemi za elektronsko učenje, Web tehnologije, Web aplikacija.

## I. UVOD

EVIDENTNO je nastojanje obrazovnih institucija da u svoju ponudu uvrste tehnološki napredna rešenja za proveravanje znanja studenata. U školama i fakultetima postoje informacioni sistemi koji su namenjeni skladištenju i prezenaciji već ocenjenih radova. Ocenjivanje je uglavnom i dalje potpuna odgovornost i obaveza nastavnika i njihovih saradnika. Sistemi za elektronsko učenje koji se sve više koriste kao podrška u obrazovnim procesima, u sebi sadrže i module/komponente za ocenjivanje. Ove aplikacije su fokusirane na procese učenja (izrada i distribuiranje nastavnih materijala, interaktivan rad, organizacija učenja) i različitih vrsta sardnje između subjekata učenja (nastavnika i studenata). Realizovani su kao samostalni sistemi, a da bi se koristili kao podsistem u IS fakulteta/škole, neizbežna je stručna pomoć i podrška autora/distributera, što unosi dodatne troškove u uvođenju sistema u operativnu upotrebu. Pored navedenog, problem ovakvog pristupa je i potencijalno narušavanje bezbednosti podataka jer se otvaraju novi nalozi, podacima se pristupa spolja, a deo funkcija administriranja više nije samo u nadležnosti institucije. Zbog navedenih razloga škole-fakulteti imaju dva izbora: ili razvijaju vlastite sisteme, ili koriste već razvijene nabavljene/kupljene sisteme izdvojene i napunjene redundantnim podacima.

Ovaj rad im za cilj da prikaže metodologiju i iskustva u razvoju sistema za ocenjivanje u postojećem IS Univerziteta Singidunum u Beogradu.

## II. UNIVERZITETSKI IS (UNIS)

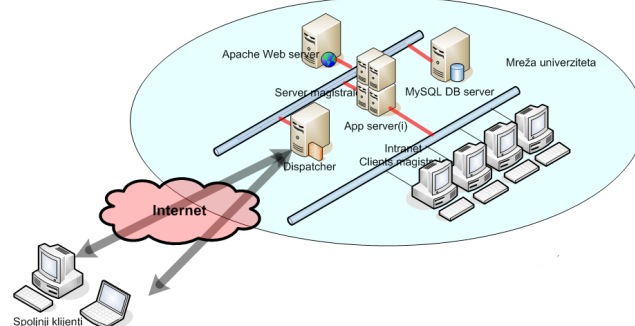
UNIS (University Network Information System) je informacioni sistem razvijen sa ciljem da podrži

<sup>1</sup>Goran Šimić, Vojna Akademija Beograd, Pavla Jurišića Šturma 1 Beograd; telefon: 381-11-3600014; e-mail: gshimic@eunet.yu

<sup>2</sup>Aleksandar Jevremović, Univerzitet Singidunum Beograd; telefon: 381-11-3093265; e-mail: ajevremovic@singidunum.ac.yu

svakodnevne aktivnosti učesnika u procesima koji se odvijaju na univerzitetu. Osnova sistema i početni skup modula su razvijeni na Univerzitetu Singidunum u Beogradu sa mogućnošću razvoja dodatnih modula za nove/specifične potrebe.

U osnovni skup modula spadaju moduli sa ugrađenom podrškom za osnovne aktivnosti korisnika sledećih kategorija korisnika: menadžment, predavači, računovodstvo, studentska služba, studenti, administratori.



SI.1. Generalna topologija sistema

Osnova UNIS sistema je izgrađena na HTTP, HTTPS, XHTML, CSS i PHP protokolima i tehnologijama a bazirana je na klijent/server i SOA arhitekturama što omogućava i korišćenje sistema putem Interneta. Kroz korišćenje apstrakcionog sloja ka bazi podataka izbegnuta je zavisnost od nekog određenog SUBP a u inicijalnoj upotrebi je iskorišćen MySQL sistem za upravljanje bazama podataka.

## III. OCENJIVANJE U SISTEMIMA ZA ELEKTRONSKO UČENJE

Postoje tri grupe sistema za elektronsko učenje [1]. To su adaptivni hipermedijalni sistemi, inteligentni tutorski sistemi i sistemi za upravljanje učenjem. Prve dve grupe su fokusirane na pomoć u fazi izučavanja i domenski su usko specijalizovane. Na primer izučavanje algebre (Cognitive Tutor) [2], ili sintakse SQL jezika (SQL Tutor) [3]. Proveravanje u ovim sistemima predstavlja deo procesa učenja i namenjeno je isključivo za profilisanje studenta radi pružanja adekvatne pomoći i individualizacije procesa učenja. Za razliku od prethodne dve grupe, sistemi za upravljanje učenjem (u daljem tekstu LMS imaju izraženu administrativnu komponentu koja treba da podrži organizovanosti i funkcionisanje kompletnog nastavnog procesa u obrazovnim institucijama. Ovo su najmnogobrojniji sistemi, koriste se više od svih ostalih sistema i njihovi reprezentativni primeri su WebCT [4] i Moodle [5].

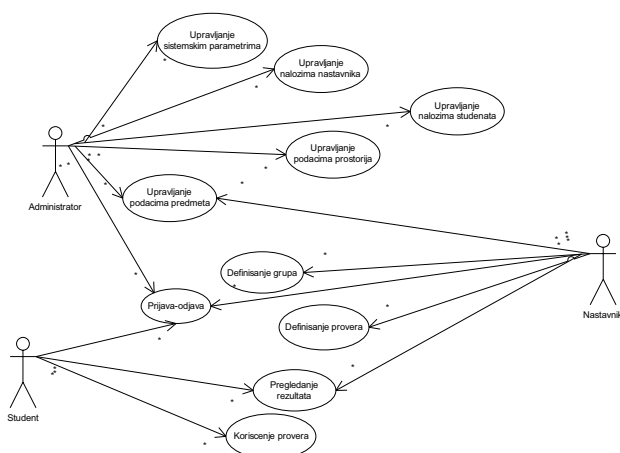
Proveravanje u LMS predstavlja posebnu funkcionalnu celinu. Fokus u dizajnu provera je da se omogući što veći izbor vrsta provera (unos odgovora/rešenja u slobodnoj formi, izbor od ponuđenih odgovora) i podrška nekog od

opšteprihvaćenih standarda za čuvanje podataka/rezultata provera. Suštinski nedostatak kod LMS je što se IS domaćina mora prilagođavati struktuiranosti LMS, ili se moraju unostiti redundantni podaci studenata, predmeta i nastavnika. Pouzdanost proveravanja i konzistentnost rezultata studenata ostvarljiva je samo u potpuno kontrolisanim uslovima.

#### IV.FUNKCIONALNOSTI SISTEMA

Osnovne funkcionalnosti u sistemu za proveru znanja podeljene su u 3 grupe: administratorska, nastavnička i studentska.

Na osnovu toga su dizajnirana 3 nezavisna softverska modula nazvana kao i funkcionalnosti koje obezbeđuju. Administratorske funkcionalnosti obezbeđuju preduslove za korišćenje sistema. Upravljanje nalogima korisnika, podacima korisničkih grupa, definisanje prostorija, predmeta i podešavanja sistemskih prametara su najvažnije funkcionalnosti ovog modula.



SI.1. Osnovni model slučajeva korišćenja sistema

Nastavničke funkcije omogućavaju kreiranje struture predmeta, kreiranje i dizajn provera, povezivanje studentskih grupa, prostorija i provera.

Osnovne studentske funkcionalnosti su da rešavaju provere i pregledaju sopstvene rezultate.

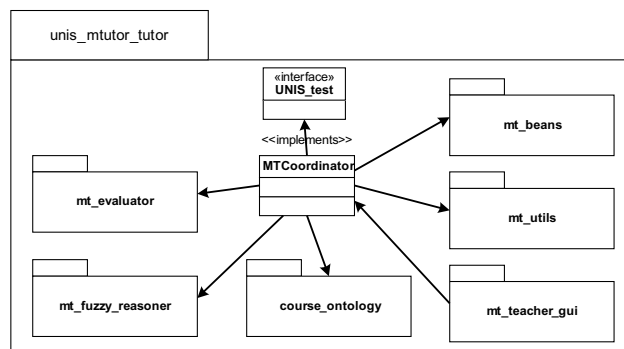
#### V.DIZAJN I IMPLEMENTACIJA SISTEMA

Jezgro *UNIS.Mtutor*-a je razvijeno na osnovu sistema *Multitutor*, samostalnog sistema za upravljanje učenjem [6,7]. U razvoju sistema *UNIS.MTutor* u okviru postojećeg IS uiverziteta pojavila su se mnoga ograničenja među kojima se ističu sledeća: tehnologija softverskog rešenja morala je biti kompatibilna sa tehnologijom IS domaćina; ograničenost pristupa na osnovne podatke studenata, predmeta i nastavnika u bazi podataka; sistem se koristi u mrežnom okruženju kojim se administrira spolja.

Zbog gore navedenih ograničenja opredelilo se za razvoj softverskih modula (pomenutih u prethodnom poglavlju), platformski nezavisnih, kojima se moglo pristupiti bilo iz desktop, ili iz Web aplikacije, u okviru lokalne (kampus) mreže na univerzitetu, ili preko Interneta.

##### A.Osnovni koncepti u modulima sistema

Nastavnički modul predstavlja najkompleksniju komponentu jer obezbeđuje najveći broj funkcija u sistemu.

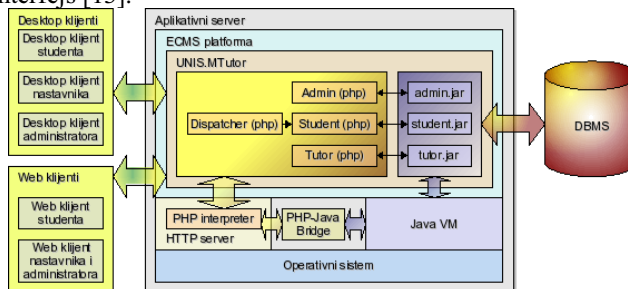


SI.2. Dizajn nastavničkog modula

Sva tri modula predstavljaju 3 nezavisna paketa koja imaju sličnu strukturu; radi pojednostavljenja komunikacije između okruženja i modula, postoji samo jedna klasa za pristup sistemu (*MTCoordinator*), čije metode predstavljaju implementaciju standardnog (API) interfejsa za komunikaciju (*UNIS\_test*), a koji je specificiran funkcionalnim zahtevima korisnika i usaglašen sa *IMS QTI* standardom [8]. U dizajnu modula je primenjen *Facade* UML obrazac [9]. Ova klasa ujedno ima ulogu eksperta i kontrolera u odnosu na sve ostale komponente sistema. Ostale klase su distribuirane u podpakovanja prema funkcionalnostima: *mt\_beans* sadrži klase za komunikaciju sa izvorima podataka (DBMS, fajlovi, internet); *course\_ontology* poseduje klase koje enkapsuliraju strukturu predmeta i ova ontologija je dizajnirana u saglasnosti sa *SCORM* standardom [10]; *mt\_evaluator* je paket čije su klase namenjene ocenjivanju studenata; za potrebe naprednog (više-kriterijumskog) ocenjivanja dizajniran je poseban set klasa smeštenih u pakovanje *mt\_fuzzy\_reasoner*; zajedno sa Java shell-ovima za rezonovanje na bazi produkcionih pravila [11] i fuzzy pravila [12]. Klase ovog modula omogućavaju rezonovanje o uspehu studenata na osnovu više kriterijuma i na osnovu nepotpunih i često neodređenih podataka. Pored sistemskog ocenjivanja, nastavnici mogu kreirati individualnu strategiju ocenjivanja posebno za svaku grupu, i/ili proveru. Pored *mt\_utils* pakovanja koje sadrži različite pomoćne klase, kao pridodat u svakom modulu, postoji paket sa klasama korisničkog interfejsa za desktop aplikaciju (*mt\_teacher\_gui*, *mt\_student\_gui*, *mt\_admin\_gui*).

##### B.Dizajn Web okruženja

U skladu sa IS domaćina razvoj je nastavljen na istim Web tehnologijama sa proširenjem koje omogućava korišćenje Java klasa. U tu svrhu je korišćen PHP-Java interfejs [13].



SI.3. Dizajn nastavničkog modula

Svakom od Java paketa odgovara istoimena PHP komponenta sa izuzetkom dispečerske komponente koja obezbeđuje horizontalnu skalabilnost sistema. Ova komponenta omogućava dodavanje neograničenog broja

aplikativnih servera u cilju raspoređivanja opterećenja pri intenzivnom korišćenju.

Klijenti isključivo komuniciraju sa UNIS.MTutor-om bez obzira da li su bazirani na desktop ili web tehnologijama. Radi povećanja pouzdanosti testiranja razvijen je poseban Web klijent - *UNIS.MTutor Web client* - baziran na JREx implementaciji Gecko engine-a [14]. Ovaj klijent garantuje kompatibilnost sa tehnologijama korišćenim za razvoj korisničkog interfejsa, ograničava navigaciju i sprečava korišćenje drugih aplikacija/dokumenata od strane studenata tokom proveravanja. Studentska komponenta na aplikacionom serveru ima mogućnost ograničavanja pristupa aplikaciji isključivo korišćenjem ovog klijenta.

## VI.HARDVERSKA PLATFORMA

Specifikacija hardverske platforme na kojoj će se koristiti *UNIS.MTutor* sistem ne predstavlja niti jednostavan niti jednodimenzionalni problem. Za određivanje optimalnih karakteristika koje noseći hardver treba da poseduje neophodno je uzeti u obzir sledeće parametre:

1. prosečan/maksimalan broj studenata koji se testira istovremeno
2. prosečan/maksimalan broj pitanja na koje student odgovara u toku jedne provere
3. prosečan/maksimalan broj odgovora po pitanju
4. prosečno vreme utrošeno za davanje odgovora

Utvrđene prosečne vrednosti upućuju na predviđeno prosečno opterećenje sistema dok maksimalne vrednosti ukazuju na vršno opterećenje koje se uglavnom javlja na početku testiranja pri učitavanju pitanja i odgovora. Na osnovu navedenih parametara mogu se odrediti sledeće karakteristike kao bitne za optimalno funkcionisanje sistema:

1. procesorska snaga dispečerskog servera
2. količina radne memorije dispečerskog servera
3. procesorska snaga aplikativnog servera
4. količina radne memorije aplikativnog servera
5. broj potrebnih aplikativnih servera
6. procesorska snaga servera baze podataka
7. količina radne memorije servera baze podataka
8. broj potrebnih servera baze podataka
9. propusna moć komunikacionih kanala između dispečerskog servera i klijentskih terminala
10. propusna moć komunikacionih kanala između aplikativnog servera i klijentskih terminala
11. propusna moć komunikacionih kanala između aplikativnog servera i servera baze podataka

## VII.EVALUACIJA

UNIS.Mtutor je do sada sistem korišćen u 1 semestru. Izvršeno je 82 provere znanja sa 1844 studenata. Sistem pohranjuje 1823 pitanja sa 6925 pripadajućih odgovora (za *single&multiple choice* tip pitanja) iz 9 predmeta. Na osnovu njih je uskladišteno 3.674 studentska rada sa 175.284 otvorena pitanja i 188.060 datih odgovora. Tokom korišćenja sistem je pokazao stabilnost i pouzdanost u radu.

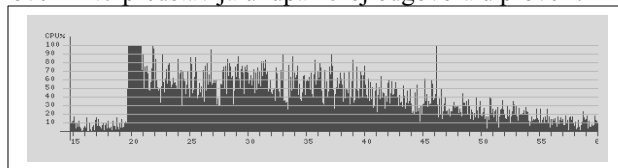
U početku eksploatacije utvrđena su *uska grla* hardverskih i softverskih komponenti sistema. Softverski problemi su se odnosili na heterogenu strukturu sistema, naročito nekompatibilnost različitih verzija korišćenih komponenti. Na primer, česte promene verzija java konektora za komunikaciju sa bazom podataka, PHP-Java

konektora i same Jave, uslovile su potrebu ažuriranja korišćenih biblioteka klasa, pa čak i izvornog koda aplikacije (zamena zastarelih metoda – *depricated methods*).

Prva korišćenja sistema realizovana su na hardverskoj platformi Intel P4 procesor 2.4GHz, 1GB RAM, 1Gb/s mrežnim linkom. Usko grlo se pojavilo prilikom pristupanja proveru od strane većeg broja studenata istovremeno (~50 studenata), kada se brzina odziva dramatično smanjila (~120s). Sistem je dizajniran da za svakog studenta u proveru vrši randomizaciju redosleda pitanja i ponuđenih odgovora. Maksimalno opterećenje procesora u trenutku pristupa proveru moglo se izraziti na sledeći način:

$$O_{\max} = N_s * N_p * \left( \frac{N_o}{N_p} + 1 \right)$$

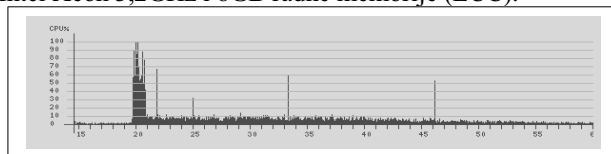
Pri čemu  $N_s$  predstavlja broj studenata,  $N_p$  je broj pitanja u proveru i  $N_o$  predstavlja ukupan broj odgovora u proveru.



Sl.4. Opterećenje aplikativnog servera sa Intel P4 2,4GHz procesorom

Na gornjoj slici predstavljeno je opterećenje serverskog procesora tokom relizacije provere nad 63 studenta koji su odgovarali na 100 pitanja i 400 odgovora. Može se primetiti da se vršno opterećenje pojavilo na početku perioda proveravanja, tj. u trenutku preuzimanja pitanja i odgovora.

U sledećoj verziji hardverske platforme, navedene komponente su zamenjene boljim: procesor sa 8 jezgara tipa Intel Xeon 3,2GHz i 8GB radne memorije (ECC).



Sl.5. Opterećenje procesora aplikativnog servera sa 8 jezgara tipa Intel Xeon 3,2GHz

Ovom promenom, uz dodatno optimizovanje koda aplikacije (ekstrakcija i enkapsulacija učestalo korišćenih metoda u statičke metode posebne klase, redukcija poziva brokerskih klasa prema BP korišćenjem unapred pripremljenih i parsiranih SQL naredbi), dobijeni su zadovoljavajući rezultati, slikovito predstavljeni na prethodnoj slici. Dostignuta je planirana brzina odziva sistema (na klijentskoj strani, u intranet okruženju) – bolja od 5s, sa preko 100 studenata, preko 500 pitanja i preko 2000 odgovora. Eventualna uska grla u budućoj eksploataciji mogu se javiti usled ograničene propusne moći komunikacionih kanala, što se može rešiti dodavanjem novih mrežnih adaptera i optimizacijom konfiguracije dispečerske komponente sistema.

## VIII.ZAKLJUČAK

U projektu *UNIS.Mtutor*, koji je relizovan za 3 meseca, trećina vremena je utrošena za analizu sistema, dizajn i implementaciju. Svo preostalo vreme iskorišćeno je za testiranje sistema u realnim uslovima i iterativno inkrementalni razvoj. Kratak vremenski rok realizacije i navedena vremenska raspodela projekta bila je ostvarljiva zahvaljujući prikupljenom velikom iskustvu u predistraživanjima i ekperetskom poznavanju tehnologija

ugrađenih u sistem. Modularna arhitektura softvera na pouzdanjoj hardvrskoj platformi i mrežnom okruženju, u procesu evaluacije je pokazala ispravnost korišćenog pristupa. Korišćenjem standardnih formata u struktuiranju podataka omogućena je interoperabilnost sa okruženjem IS domaćina i srodnim sistemima.

U sledećoj verziji *UNIS.Mtutor*-a planirana su dva unapređenja; proširivanje studentskog modela, čime bi se u ocenjivanju obuhvatili i drugi podaci o studentima (posećenost nastavi, podaci seminarara, aktivnost studenta) i postigla veća pouzdanost sistema (npr. otkrivanje neregularnosti i/ili kontradiktornosti). Druga novina je ugradnja *problem generatora* sa pratećim komponentama koje bi omogućile automatsko generisanje pitanja/zadataka i rešenja. Ocenjivanje bi se omogućilo upoređivanjem studentskog rešenja sa rešenjem koje daje generator. Prototip ove komponente je već implementiran i testiran za zadatke iz oblasti finansija. Postiguti rezultati ukazuju na opravdanost njenog daljeg razvoja.

#### LITERATURA

- [1] P. Brusilovsky, *A Distributed Architecture for Adaptive and Intelligent Learning Management Systems*, In Proceedings of the AIED 2003 Workshop Towards Intelligent Learning Management Systems, Sydney, 2003, pp. 5-13.
- [2] S.Ritter, *PAT Online: A Model-Tracing Tutor on the World-Wide Web*, Proc. from Workshop about Intelligent Educational Systems on the World Wide Web, Kobe, Japan, 1997, pp.11-17.
- [3] Intelligent Tutoring System - *SQL Tutor*, Available: <http://www.cosc.canterbury.ac.nz/~tanja/sql-tut.html>.
- [4] *WebCT*, Available: <http://www.webct.com>.
- [5] C.Brayan, *Introduction to Moodle*, preuzeto sa <http://www.moodle.org>, 2006.
- [6] G.Šimić, V.Devedžić, *Building an intelligent system using modern Internet technologies*, Journal Expert Systems with Applications, Elsevier, 2003., Vol. 25, pp. 231-246.
- [7] G.Šimić, *The Multi-course Tutoring System Design*, Journal Computer Science and Information Systems, Vol.1, February 2004., pp. 141-155.
- [8] *IMS Question&Test Interoperability IMS QTI* Available: <http://www.imslobal.org>.
- [9] C. Larman, *Applying UML and Patterns*, Prentice-Hall, Englewood Cliffs, NJ, 1999.
- [10] Standard za metapodatke *SCORM* Available: <http://www.adlnet.org>.
- [11] E.Friendman, *Jess in Action - Rule-Based Systems in Java*, Sandia National Laboratories Livermore, CA, 2003.
- [12] R.Ochard, *NRC FuzzyJ Toolkit for the Java(tm) Platform User's Guide* (version 1.9), Institute of Information Technology National Research Council of Canada, 2006.
- [13] PHP-Java bridge, Available: <http://php-java-bridge.sourceforge.net/>
- [14] Web pretraživačka komponenta zasnovana na Java programskom jeziku *Jrex Gecko* Available: <http://jrex.mozdev.org/index.html>.
- [15] *UNIS.Mtutor* Available: <http://tool.singidunum.ac.yu/demo> (URL)

#### ABSTRACT

This paper is about development of *UNIS.Mtutor* assessment tool. There are several e-learning systems. Disadvantages of these systems are: narrow domain specialisation, the small ability of adopting to actual needs of educational institutions and, the limited possibilities of assessment. The above characteristics as well as the previous researches, results and experience represents the base motives for development of *UNIS.Mtutor* inside the University Information System. The main content of the text is represented with the most important design details and evaluated experience.

#### The Development of Assessment Tool Inside the