Servisno-orijentisani pristup implementaciji i testiranju adaptivnog e-learning sistema

Ljubomir Lazić
METROPOLITAN Univerzitet, Beograd
ljubomir.lazic@metropolitan.ac.rs

Nebojša Gavrilović
METROPOLITAN Univerzitet, Beograd
nebojsa.gavrilovic@metropolitan.ac.rs

Sadržaj—**U ovom radu se opisuje jedno rešenje e-learning sistema koje je zasnovano na veb servisima tj. implementacija na bazi servisno-orijentisane softverske arhitekture. Predloženo rešenje je dizajnirano u skladu sa strategijom za personalizaciju procesa učenja kroz: filtriranje na bazi karakterističnih metapodataka za svaki objekat učenja, personalzaciju kroz izbor objekata učenja i osiguranje rezultata provere znanja studenta. Osiguranje kvaliteta e-learning sistema je postignuto primenom adekvatnih tehnika testiranja veb servisa i testova za proveru znanja studenta tokom personalizovanog procesa učenja.**

Ključne reči - e-learning; personalizovano učenje; LAMS; SOA; SOA testiranje; Java grader; BIT softver

I. UVOD

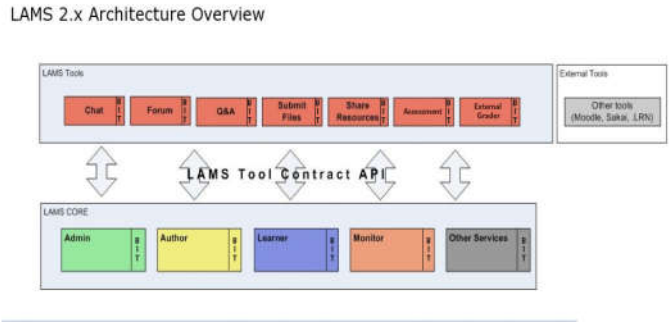
Nastavni materijali koji se koriste u personalizovanim procesima učenja moraju biti u formi objekata učenja - OU (malih jasnih celina) koji sadrže karakteristične metapodatke [1,2]. Težnja ka podeli nastavnih materijala na objekte učenja omogućava ponovno korišćenje (*reuse*) u različitim procesima učenja. Definisanje karakterističnih metapodatka za svaki objekat učenja omogućava kasniju pretragu u okviru repozitorijuma objekata učenja unutar sistema. Repozitorijum objekata učenja predstavlja deo sistema u kome se vrši skladištenje svih objekata učenja. Profesor, autor kursa, kreira nastavni materijal pisanjem objekata učenja i formiranjem procesa učenja od napisanih objekata. Pored kreiranja (pisanje objekata učenja od početka) autor kursa može koristiti postojeće objekte učenja napisane od strane drugih autora. Definisanjem pretrage repozitorijuma objekata učenja korišćenjem karakterističnih metapodataka (ključnih reči, klasifikacije, očekivanog nivoa znanja) autor može pronaći objekat učenja i uneti ga u svoj novokreirani proces učenja. Na taj način izbegnuto je pisanje već postojećeg sadržaja nastavnog materijala od početka.

Tokom procesa učenja u sistemu je potrebno izvršiti: filtriranje, personalizaciju i ocenjivanje. Filtriranje podrazumeva definisanje nivoa znanja studenta i određivanje daljeg toka kroz proces učenja. U te svrhe razvijeno je više softverskih komponenti za ocenjivanje nivoa znanja studenta, kao što je ovde opisani spoljni ocenjivač za Java programski jezik (*Java grader*). Korišćene su tehnologije i proces integrisanja sa LAMS-om (*Learning Activity Management System*) na bazi servisno-orijentisane softverske arhitekture [1,3]. Osiguranje kvaliteta e-learning sistema je postignuto primenom adekvatnih tehnika testiranja veb servisa (primenom BIT tehnika i softverskih komponenti u cilju testiranja tj. *Built-In-Test Software*) i testova za proveru znanja studenta tokom personalizovanog procesa učenja. Primenjena je "Cause-Effect" tehnika u testiranju e-learning aplikacije, jer je pogodna za automatizovanje kao i činjenica da se test slučajevi izvlače iz softverske specifikacije date prirodnim jezikom [2,4].

II. VEB SERVISI U LAMS ADAPTIVNOM PROCESU UČENJA

A. Opis predloženog LAMS-SOA rešenja

Servisno-orijentisana arhitektura (*Service-Oriented Architecture, SOA*) predstavlja iskorak u oblasti razvoja softvera značajan koliko i prelazak sa proceduralnog na objektno-orijentisani način razmišljanja, analize, dizajna i razvoja. Iako deli mnoge karakteristike sa OO konceptima (ponovno iskorišćenje komponenti, ugovaranje interfejsa itd.), SOA uvodi nove potencijale (pre svega kod izgradnje mrežnih informacionih sistema) koji je ističu kao ultimativnu platformu savremenih distribuiranih sistema za obradu podataka. Servisno-orijentisana arhitektura se može smatrati i primenom proverenih OO koncepata u cilju odgovora na savremene potrebe za distribuiranom obradom podataka i komunikacijom između raznorodnih softverskih sistema [5]. Primena SOA arhitekture u sistemima za učenje na daljinu nije novost. Veliki broj razvojnih timova koristi SOA za implementiranje funkcionalnosti u okviru sistema [5]. Pored korisničkih zahteva, predstavljeni su i poslovni procesi koji se trenutno odvijaju korišćenjem LAMS-a (Slika 1) i informacionog sistema Univerziteta Metropolitan (ISUM).



Slika 1. LAMS 2.x arhitekura sa implementiranim BIT omotačem komponenti

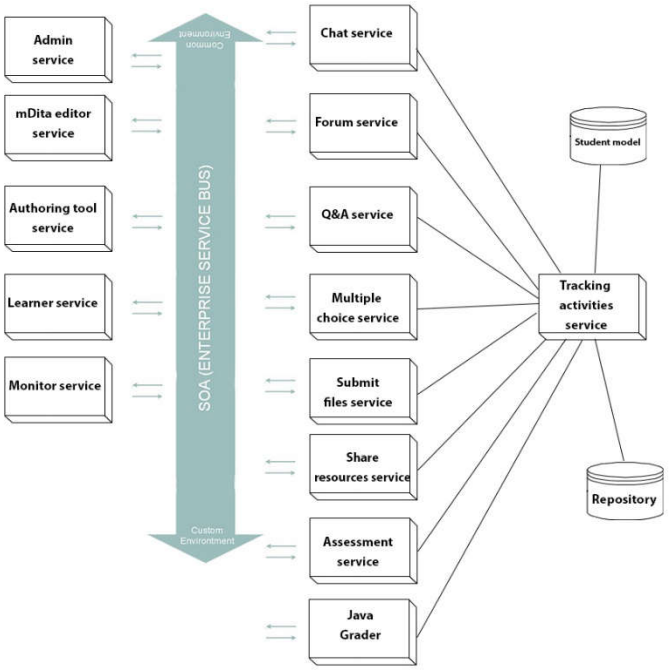
Cilj je izvršiti integraciju sistema i omogućiti razmenu podataka između LAMS i ISUM sistema tako da korisnici imaju sve podatke sa sistema za učenje na daljinu unutar

informacionog sistema Univerziteta. Specifičnosti rešenja koje je predloženo u ovom radu se može primetiti na osnovu analize LAMS-SOA arhitekture koja je prikazana na sledećoj slici 2.

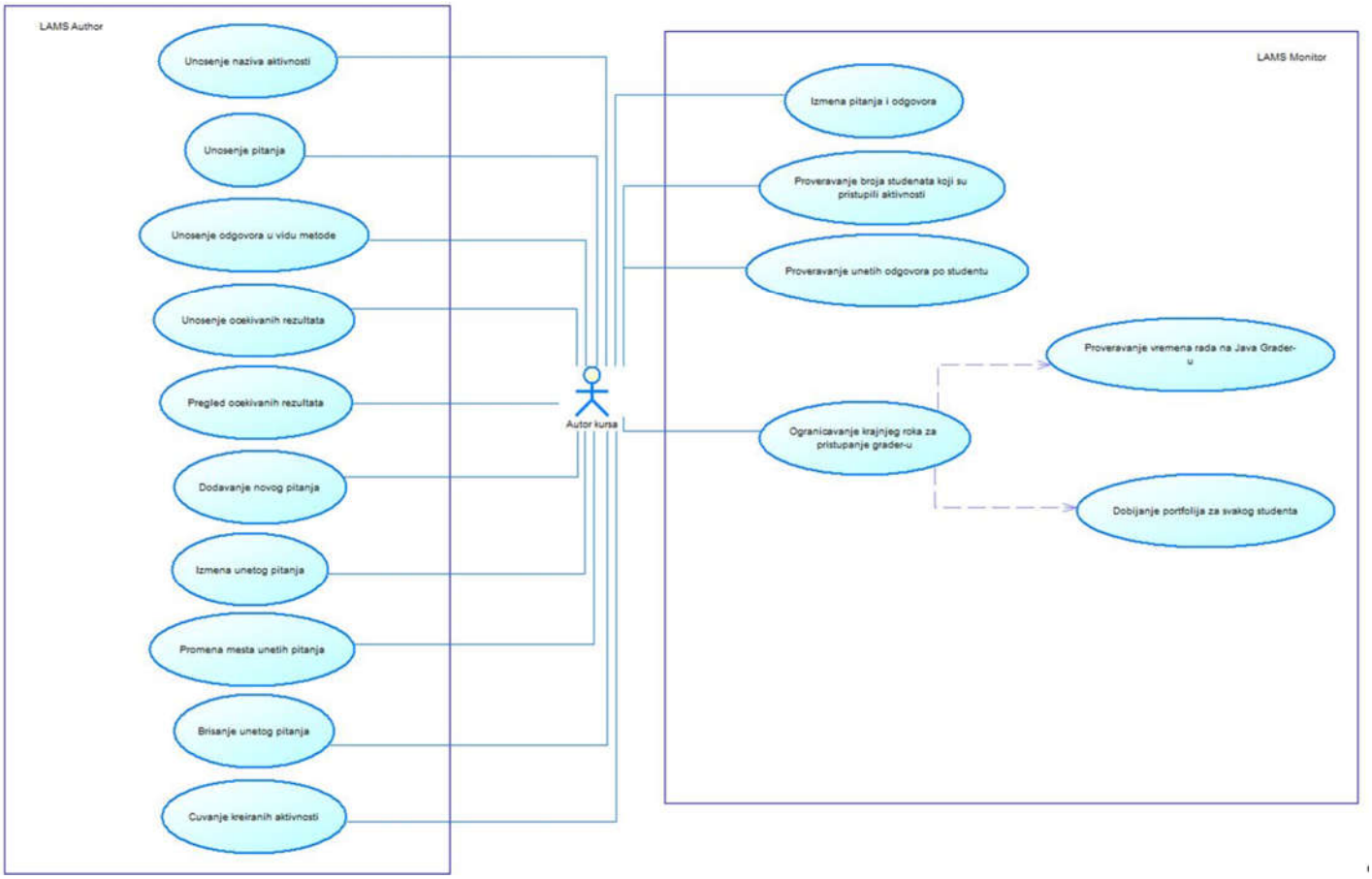
B. *LAMS – SOA servisi*

Sa slike 2 se vidi komunikacija između servisa koja se ostvaruje tako što *Admin service* treba da ostvari komunikaciju sa svim servisima prikazanim na desnoj strani od BUS magistrale a to su:1. *Chat service*; 2. *Forum service*; 3. *Q&A service* (jedan tip testa provere znanja studenta); 4. *Multiple Choice service* (drugi tip testa provere znanja studenta); 5. *Submit files service*; 6. *Share resources service*; i 7. *Assessment service* (složeni tip ocene znanja studenta).

Na slici 3 su prikazani slučajevi upotrebe autora i monitora kursa za LAMS- SOA servise.



Slika 2. LAMS- SOA servisi

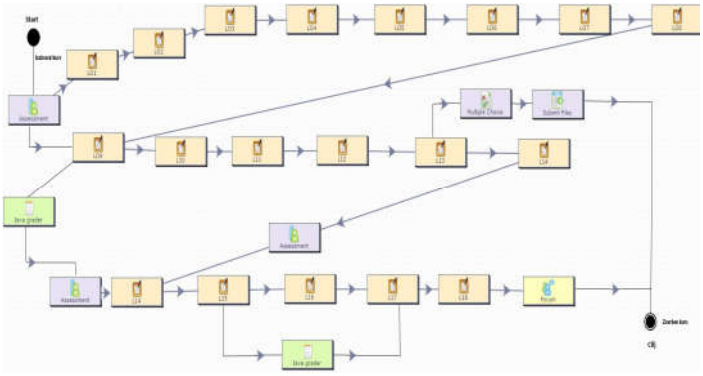


Slika 3. Use Case dijagram - autor kursa i monitor kursa

Takođe, *learner* i *monitor service* treba da ostvari komunikaciju sa istim servisima kao i *administrator service*. Na slici 4 je prikazano da student kroz *Learner servis* unosi odgovor na pitanje dok od strane sistema i servisa sa desne strane dobija povratnu informaciju u vidu ocene svog unetog odgovora.

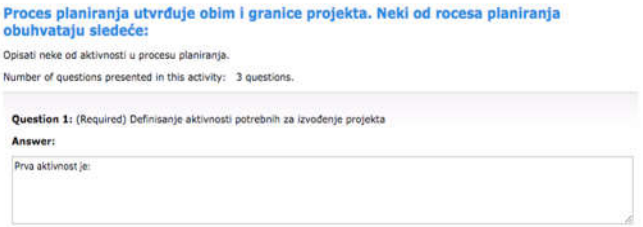
III. LAMS SOA REŠENJE PERSONALIZOVANOG E-LEARNING SISTEMA

Proces adaptivnog (personalizivanog) učenja se ogleda u krojenju složene sekvence lekcija na osnovu implementiranih aktivnosti ocenjivanja (u tačkama za ocenjivanje - *assessment gate*) potrebnih predznanja studenta za odabrani kurs predmeta tj stepena pripremljenosti koji je prikazan na slici 3. Funkcionisanje sistema određeno je tako što student korišćenjem Learner servisa pristupa LAMS-u. Funkcionisanje sistema određeno je tako što student korišćenjem Learner servisa pristupa LAMS-u*assessment gate*) potrebnih predznanja studenta za odabrani kurs predmeta tj stepena pripremljenosti koji je prikazan na slici 3. Funkcionisanje sistema određeno je tako što student korišćenjem *Learner servisa* pristupa LAMS-u.



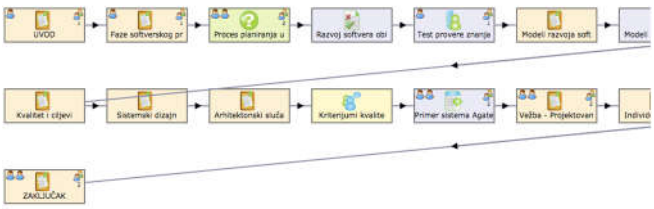
Slika 3. Proces personalizovanog učenja za izabrani kurs predmeta

Kada student otvori lekciju, zavisno od LAMS aktivnosti (servisa) koje se nalaze u lekciji *student service* vrši komunikaciju. Ukoliko, primera radi, u lekciji postoji Q&A aktivnost, *student service* će uspostaviti komunikaciju sa Q&A servisom.



Slika 4. *Learner service* - unos odgovora

Razmena informacija između Q&A servisa tiče se podataka da li je student odgovorio na postavljeno pitanje u Q&A aktivnosti, ukoliko jeste šalje se podatak u *student service* i student kroz servis dobija povratnu informaciju da je odgovorio na pitanje.



Slika 5. *Monitor service* - pregled aktivnosti

Nakon slanja informacije studentu da je odgovorio na pitanje, sledeći korak je slanje profesoru kroz *monitor service* da je student uneo odgovor na aktivnost Q&A. Na slici NEMA BROJ SLIKE je prikazan proces učenja i treća aktivnost je Q&A gde student uneo odgovor (prikazan je u vidu figure na aktivnosti koja predstavlja studenta). Profesor, kroz *monitor service* može pristupiti i proveriti odgovor studenta unutar Q&A service-a.

Razlika između *monitor i administrator service*-a je u tome što administrator service ima mogućnost brisanja aktivnosti i lekcije dok se kroz monitor service vrši pregled unetog sadržaja. Profesor unosi ocenu na rad studenta unutar aktivnosti ako to nije automatski urađeno od strane sistema. Na slici 5 je prikazano da profesor kroz Monitor service, na osnovu dobijenih odgovora studenta kroz određene servise, unosi ocenu rada studenta.

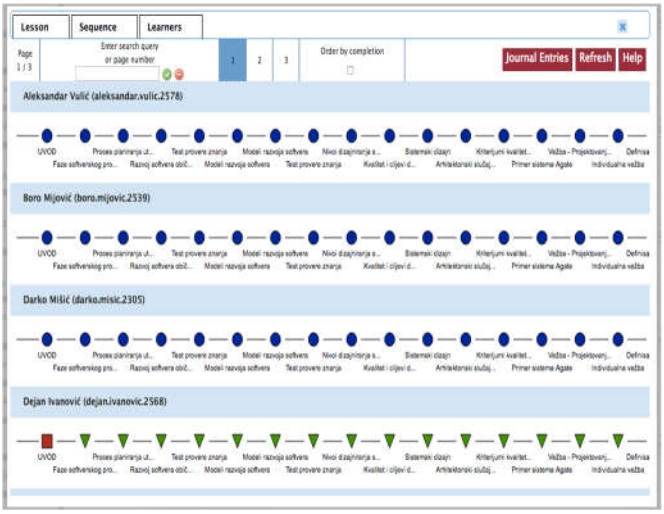
Prethodno opisani LAMS servisi (*Forum service, Multiple Choice service, Submit files service, Share resources service i Assessment service*) funkcionišu po istom principu kao Q&A service i komuniciraju sa *Monitor* i *Administrator service*.

Naš implementirani *mDITA service* je unutar LAMS-a i koristi se za prezentovanje nastavnih materijala u vidu HTML-a i PDF verzija. Ovaj service služi da napisane lekcije pretvori u format koji odgovara prikazu na LAMS-u. Kroz ovaj service moguće je generisati proces učenja ili PDF verziju lekcije. Na istoj slici 2 je prikazan tok podataka od strane *mDita service*-a u vidu slanja nastavnog materijala sa dodatnim aktivnostima dok se kao odgovor dobijaju postavljene aktivnosti unutar Sistema (Slika 5). Time se kompletira proces učenja korišćenjem nastavnih materijala i dodatnih aktivnosti. Komunikacija je ostvarena u vidu linkova koji se postavljaju na LAMS, dok su materijali na *mDita serveru*. Takođe, ovaj service je povezan i sa aktivnostima unutar LAMS-a koristeći glavni *LAMS interface* i *Authoring tool* service. Poslednji service sa leve strane o kome će biti reči je *Authoring tool service*. Korišćenje ovog servisa omogućeno je administratorima i monitorima unutar sistema pa je iz tog razloga omogućena komunikacija sa *Admin* i *Monitor service*-om. *Authoring service* služi za kreiranje procesa učenja na LAMS-u i dodavanje dodatnih aktivnosti. Kada se kroz *mDita service* izvrši generisanje nastavnog materijala i linkovi se postave na LAMS, dodatno modifikovanje i podešavanje procesa učenja se vrši kroz *Authoring service*. Korisnik (administrator ili profesor) pristupaju servisu i vrše modelovanje procesa učenja.

Na slici 2 je prikazan *Authoring service* i razmena podataka ide tako što podaci koji se šalju su procesi učenja (nastavni materijali u obliku objekata učenja i dodatne LAMS aktivnosti) dok je kao povratna informacija od strane sistema informacije o dodatnim aktivnostima dodatih u proces učenja od strane autora nastavnog materijala.

Kroz *Authoring tool service* interfejs moguće je dodati aktivnosti koje su navedene na slici sa desne strane BUS magistrale a tiču se dodatnih LAMS aktivnosti i provere

znanja studenta u lekciji. Na taj način korisniku je omogućeno da kreira dodatne aktivnosti, određuje pravila unutar njih (vreme za koje je potrebno odgovoriti, datum početka, broj poena).



Slika 6. *Tracking activities service* - provera rada studenata

Iz *Tracking activities service*-a podaci odlaze u student model i repository. Repository služi za skladištenje svih informacija, kako o aktivnostima studenata tako i profesora, proverenim aktivnostima, ocenama koje su unete. Student model služi da bi se studentu na osnovu zabeleženih informacija u lekciji i aktivnostima automatski od sistema dodelila naredna lekcija. Ovaj deo je u fazi razvoja i za sada se samo upisuju podaci od studenata dok će u narednoj iteraciji biti omogućeno i povezivanje sa drugim lekcijama sličnog nivoa. U okviru sistema moguće je generisanje različitih tipova izveštaja korišćenjem različitih podataka unutar sistema. Na ovaj način profesori i administratori mogu proveriti aktivnosti studenata i procenat otvorenih lekcija na sistemu.

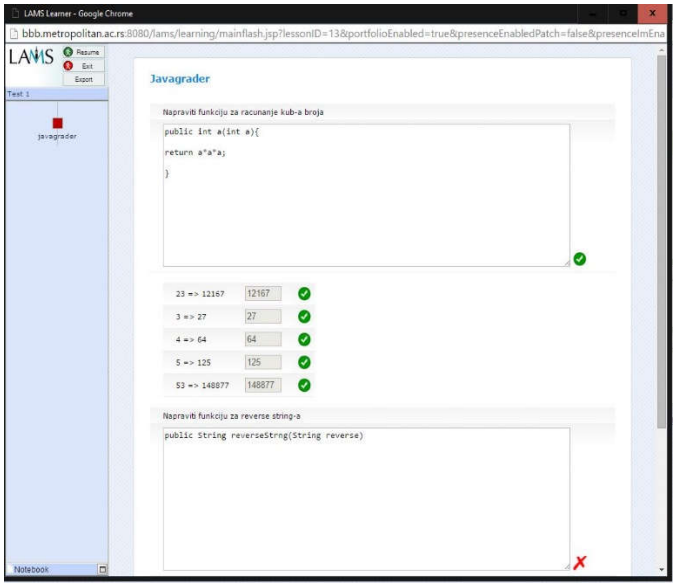
A. Novi LAMS servisi - Java grader

Korišćenjem SOA pored servisa koji već postoje na LAMS-u (Slika 2) moguće je izvršiti dodavanje i novih servisa koji će služiti za proveru znanja studenata iz oblasti programiranja. U ovom radu daje se opis jednog razvijenog dodatnog servisa koji vrši proveru Java programskog kôda direktno na LAMS-u. Servis se zove *Java grader* i kroz sistemsku SOA povezan je sa:

- Admin service
- mDita editor service
- Authoring tool service
- Learner service
- Monitor service
- Tracking activity service

Unos *Java grader* aktivnosti se vrši kroz *Admin* ili *Monitor service* korišćenjem *Authoring tool* i *Java grader service*-a. Student pristupa aktivnosti kroz *Learner service* dok profesor može vršiti proveru takođe kroz *Monitor service*. Svi podaci o radu studenta u *Java Grader service*-u skladište se u *Tracking activity service* (slika 6).

Razvojem i primenom *Java grader service*-a omogućeno je dodatno unapređenje i prilagođavanje LAMS-a za sve programske jezike koji se na fakultetu izučavaju (C++, C#, HTML i drugi). Potrebna dokumentacija softverske arhitekture je kreirana i dostupna za dalje unapređenje i ostvarivanje dodatne komunikacije između servisa sistema. Na sledećoj slici je dat jednostavan primer automatske (softverske, spoljašnje) provere metode stepenovanja celobrojne vrednosti na kub, skup test slučajeva za proveru korektnosti Java kôda i rezultat o ispravnosti Java kôda koji je student napisao.



Slika 7. Validacija odgovora studenta pomoću komponente *Java Grader*

Na ovaj način omogućeno je korisniku, studentu, mogućnost provere tačnosti metode na osnovu parametara unetih od strane autora kursa. Student će nakon popunjavanja forme imati informaciju o unetom odgovoru i proveru tačnosti na osnovu parametara:

Ulazi

• Odabir aktivnosti "Java grader"

• Uneta metoda Java programskog jezika

• Sintaksno proverena uneta metoda

• Definisani parametri za proveravanje metode

Izlazi

• Prikazani rezultati metode na osnovu unetih parametara

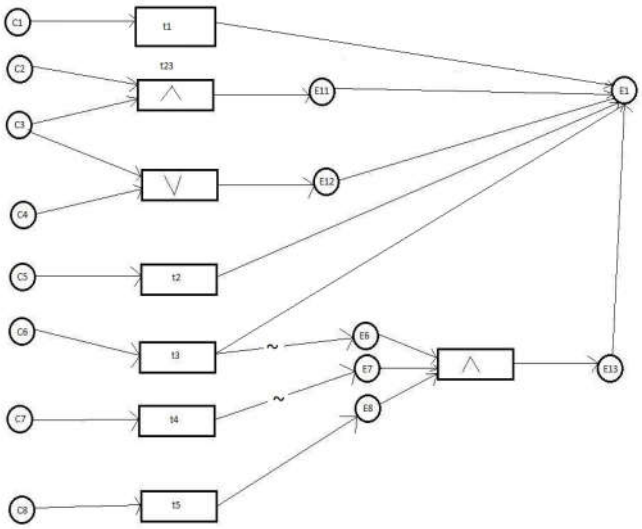
• Informacija da li je odgovor na pitanje tačan ili netačan

Za generisanja test slučajeva *Java Grader* koristi objekte učenja iz drugog kursa koji se bavi jediničnim testiranjem Java programa npr. JUnit.

B. Složeni scenario ocenjivanja studenta - Assessment service

Za složeni tip ocene znanja studenta - *Assessment service* , autor kursa studentu nudi više formi pitanja u vidu složenog scenarija koji je detaljno u našem radu [2], a u ovom radu se

daje samo ilustracija primenjene „Cause-Effect“ analize i kombinatornog testiranja. U cilju prilagođavanja narednih aktivnosti učenja na e-learning sistemu svakom studentu se nudi serija pitanja različitog tipa kao što su:C1, C2, C3, C4,C5 tipa pitanja sa dva ponuđena odgovora (*True/False*), C6, C7 i C8 pitanja sa više ponuđenih odgovora koji se boduju. Kao rezultat uspešnosti studentu se nude dodatna pitanja koji mu omogućavaju da zaradi ekstra (bonus) poene, kao i kombinacija međuzavisnih pitanja da bi se izbegli slučajevi slučajnog ispravnog odgovora prikazanom na sledećoj slici:.



Slika 8. Konačni Cause-Effect graf za opisani scenario testiranja personalizovanog učenja

Obezbeđenje korektnosti ovakvog složenog kombinovanog sistema pitanja i odgovora zahteva veoma detaljno i inteligentno testiranje. Kao efikasan način testiranja predlaže se primena kombinacije dve tehnike funkcionalnog testiranja softvera – „*Cause-Effect*“ analize i kombinatornog testiranja poznatog na primeru konkretnog složenog scenarija pitanja (C1 do C8) i odgovora na sledeći način:

C1 je prvo tip pitanje gde tačnim odgovorom (true) student prolazi i dobija skor **E1**.

C2 je drugo true/false tip pitanje gde kombinacijom sa pitanjem **C3** koje je takođe true/false i tačnim odgovorom dobija skor **E11**.

C3 je true/false tip pitanja i u kombinaciji sa **C4** (četvrto true/false pitanje) prikazuje studentu rezultat **E12**.

C5 je true/false tip pitanja koji u transformaciji tačnim odgovorom daje studentu rezultat **E1**.

C6 je multiple choice tip pitanja gde 100% (ako je **E6=2** ima stanje true u svim ostalim slučajevima obeležavanja odgovora je false) u kombinaciji sa **C7** 100% (ako je **E7=3** ima stanje true u svim ostalim slučajevima obeležavanja odgovora je false) i u kombinaciji sa **C8** dodatnim multiple choice potpitanjem (ako je **E8=3** ima stanje true u svim ostalim slučajevima obeležavanja odgovora je false) vodi studenta ka dodatnim poenima za zalaganje (dodatnih 50%) i dobijanju **E13** rezultata uz dodatnu proveru znanja.

Kroz multiple choice student može netačno odgovoriti na pitanje **C6** (na slici 8 obeležen simbolom negacije ~) ali svoj rezultat može popraviti na pitanju **C7** koje dolazi nakon pitanja **C6**. Dodatni poeni za zalaganje mogu se ostvariti na pitanju **C8** koje u ukupnom zbiru i rezultatu **E13** donose broj poena. Na taj način student ima mogućnost da ispravi svoju grešku u procesu učenja kroz dodatni set ponuđenih pitanja.

IV. KOMPOZITNI PROCES TESTIRANJA – LAMS SOA PERSONALIZOVANOG E-LEARNING SISTEMA

Da bi se osigurala brža evolucija, rekonfiguracija veb servisa i raspoređivanje servisa LAMS - SOA personalizovanog e-learning sistema, dizajniran je i implementiran okvir za automatizovano asembliranje i onlajn testiranje servisa i procesa učenja. Osiguranje kvaliteta e-learning sistema je postignuto primenom kompozitnog procesa na bazi tehnika testiranja modela veb servisa (primenom BIT tehnika i softverskih komponenti u cilju testiranja tj. *Built-In-Test Software*) i testova za proveru znanja studenta tokom personalizovanog procesa učenja. Novi pristup se ogleda u stalnom (online) testiranju radi otkrivanja grešaka stanja personalizovanog procesa učenja usled komponovanja servisa, kao što je na slici 9 prikazano, kroz: filtriranje na bazi karakterističnih metapodataka za svaki objekat učenja, personalzaciju kroz izbor objekata učenja i osiguranje rezultata provere znanja studenta. Ovaj pristup je adaptiran prema projektnom rešenju ARTOSC Composites (*Runtime Testability and Testing for Service-Oriented Architectures - SOA*) [6] koji takođe primenjuje BIT softver. Da bi se obezbedili uslovi za testiranje česte rekonfiguracije servisa LAMS - SOA arhitekture, dizajnirana je i implementirana BIT infrastruktura (videte sliku 1 i 2) koja omogućava polu- ili potpunu automatizaciju procesa testiranja [4,5]. To je rešeno primenom BIT test okvira (slika 10) na bazi:

- ➔ Automatskog ili poluautomatskog generisanja BIT omotača (wrapper)
- ➔ Automatskog ili poluautomatskog generisanja test drajvera za svaku LAMS-SOA komponentu
- ➔ Automatskog ili poluautomatskog generisanja test slučajeva

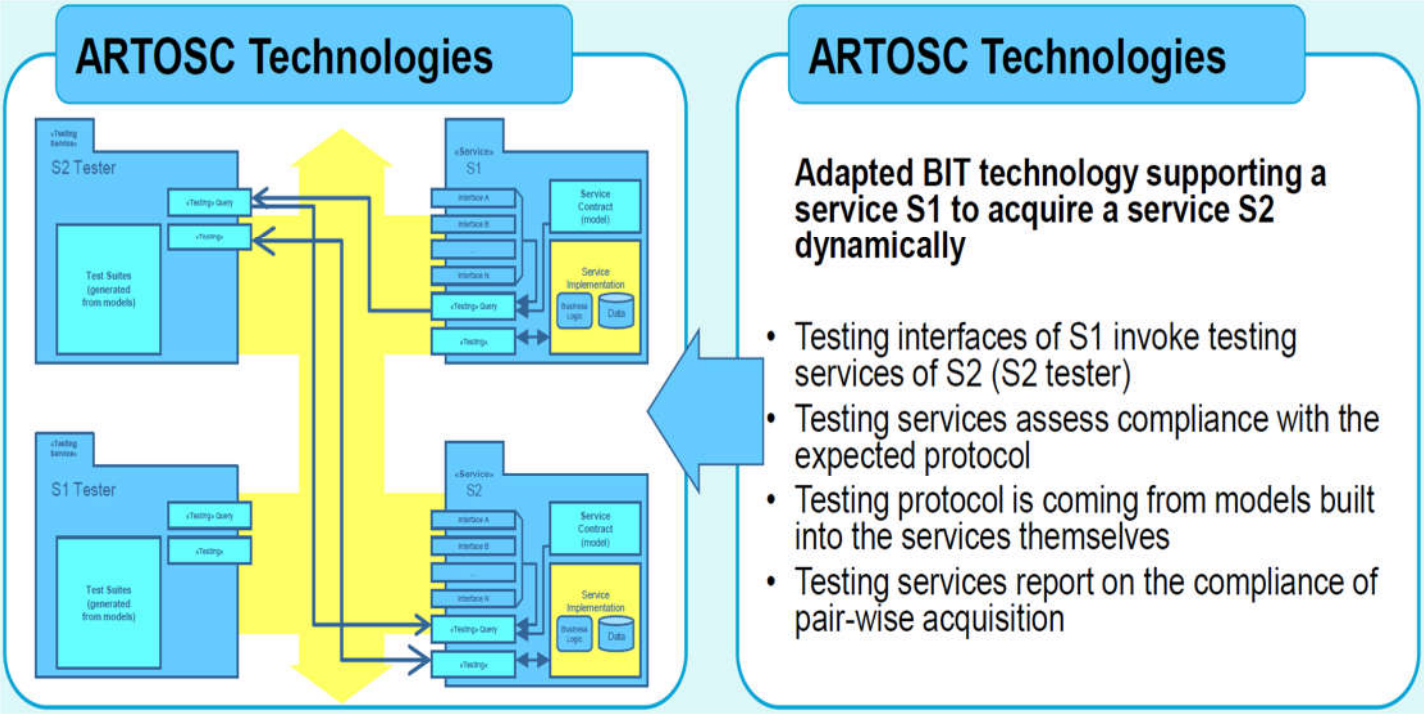
V. ZAKLJUČAK

Predloženo rešenje LAMS-SOA obezbeđuje, kvalitetan sistem personalizovanog učenja u kome je potrebno izvršiti: filtriranje, personalizaciju i ocenjivanje. U te svrhe razvijeno je više softverskih komponenti za ocenjivanje nivoa znanja studenta, kao što je ovde opisani spoljni ocenjivač za Java programski jezik (*Java grader*).

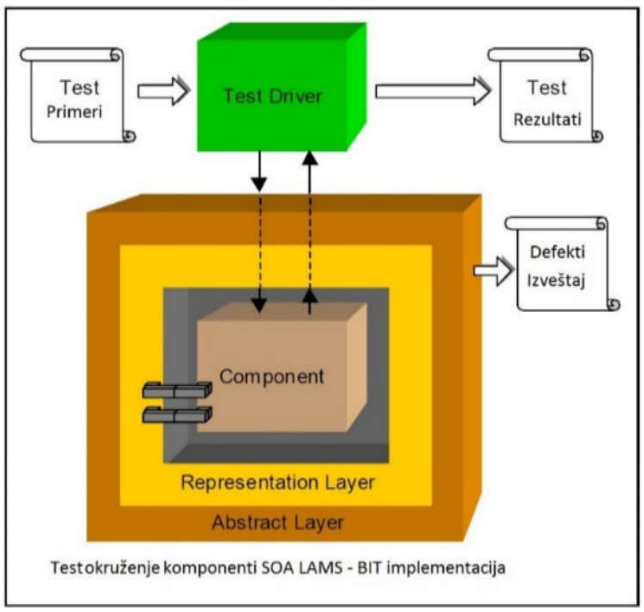
Osiguranje kvaliteta e-learning sistema je postignuto primenom kompozitnog procesa na bazi tehnika testiranja modela veb servisa primenom BIT tehnika i softverskih komponenti koje generišu složene testove za proveru znanja. Implementacijom posebnog *Assessment service*, autor kursa studentu nudi više formi pitanja u vidu složenog scenarija koji je u ovom radu opisan kao ilustracija primenjene „Cause-Effect“ analize i kombinatornog testiranja složenog načina vrednovanja znanja studenta.

ZAHVALNICA

Ovaj rad je realizivan u sklopu istraživanja koje delimično finansira Ministarstvo prosvete i Tehnološkog razvoja, Srbija na projektu 'Softversko okruženje za optimalni razvoj kvalitetnog softvera', Projekat br.TR-35026.



Slika 9. Testriranje servisa, protokola i interfejsa sa BIT softverom



Slika 10. Testno okruženje na nivou softverskih komponenti

LITERATURA

[1] N. Gavrilović, *Software for the external grader and the analysis and implementation of the learning process in LAMS*, METROPOLITAN University, master's Thesis, 2016.

[2] N. Gavrilović, Lj. Lazić, KNOWLEDGE ASSESSMENT USING CAUSE-EFFECT GRAPHING METHODS, The Seventh International Conference on eLearning (eLearning-2016), Belgrade, Serbia, 22 – 23 September 2016.

[3] D. Vassileva and B. Bontchev, A Service-Oriented Approach For Implementing An Adaptation Engine For E-Learning, The 5th International Scietific Conference, eLearning and Software for Education, Bucharest, April 09-10, 209.

[4] M. Jaffar-ur Rehman et. All, Testing software components for integration: a survey of issues and techniques, *Softw. Test. Verif. Reliab.* 2007; 17:95–133.

[5] Jean-Jacques Dubray, Composite Software Construction, Understanding SOA in the Context of a Programming Model, © 2007 C4Media Inc, 2007.

[6] http://swerl.tudelft.nl/bin/view/ARTOSC/

ABSTRACT

This paper dials with a model for adaptation of the learner assessment and the content of one learning system. The model is based on Computer Adaptive Test Theory (CAT) and organization of the learning domains, applying knowledge assessment using cause-effect graphing methods and external BIT software.

A SERVICE-ORIENTED APPROACH FOR IMPLEENTING AND TESTING E-LEARNING SYSTEM

Lj. Lazić, N. Gavrilović