

AUTOMATIZOVANO TESTIRANJE KAO METOD UČENJA

Armin Berecki¹, Sanja Mihok²

^{1,2}Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, Srbija

¹armin@uns.ac.rs, ²sanjao@uns.ac.rs

Kratak sadržaj: Testiranje znanja tokom trajanja kursa pokazalo se odlično za akumuliranje znanja. Moderni načini testiranja zahtevaju razvijanje softvera kao medija za samostalno testiranje. Jedan od glavnih izazova kod korišćenja softverskih paketa je povratna informacija. Razvojem odgovarajućeg softvera omogućava se automatizovano samotestiranje.

Ključne reči: Testiranje, Softver, Povratna informacija

Abstract: Testing during the course proved to be an excellent tool for accumulating knowledge. Modern methods of testing require development of the software as a medium for self-testing. One of the main challenges in using software packages is proper feedback. By developing appropriate software, automated self-testing can be possible.

Key Words: Testing, Software, Feedback

1. UVOD

Testiranje znanja se u konvencionalnim kursevima uglavnom odvija nakon pređenog dela gradiva ili na kraju samog kursa. Međutim, testiranjem tokom procesa prenošenja znanja pokazalo se odlično tokom trajanja kursa. Tema ovog rada je upravo testiranje tokom procesa učenja koje se može automatizovati.

Jedna od prednosti testiranja tokom kursa je koncept koji se naziva „Počni sa neuspehom“. Ova metoda pokazala se dobra na početku kursa zbog samostalnog prepoznavanja nivoa znanja studenta. Pažljivo sastavljen test na početku kursa pokreće razmišljanje i korišćenje predhodnog znanja. Takođe može pružiti studentima format zadatka i ključne informacije na koje treba obratiti pažnju tokom sadržaja koji predstoji. Naslućivanje rešenja pokreće razmišljanje koje nije direktno pamćenje informacija. Testovi trebaju biti zahtevni što se tiče nivoa znanja. Takođe moraju biti i poželjni kako bi držali pažnju.

2. SOFTVERSKO REŠENJE

Postojeća softverska rešenja daju određene načine testiranja koja nisu adekvatna u kursevima gde student treba samostalno da reši zadatak ili projetni rad koristeći različite koncepte naučene tokom kursa i tokom predhodnih kurseva. Ovi načini se uglavnom baziraju na:

- Pitanje – odgovor
- Odgovor sa „da“ ili „ne“
- Više ponuđenih odgovora
- „Drag and drop“
- Spajanje ili uvrštavanje reči i sl.

Ono što bi najviše odgovaralo testiranju vrste gradiva u kojoj je potrebno samostalno rešavanje zadatka je upisivanje rešenja u vidu matematičkih izraza i/ili brojnih vrednosti sa jedinicama mera. Ovakav softver je prilično zahtevno napisati i teško je da softver dobije „user-input“ u vidu matematičkih izraza.

Rešenje interfejsa dato je na slici 1. U gornjem levom uglu korisnik može odabrati neku od veličina koje unosi u prvo polje, dok u gornjem desnom uglu takođe treba izabrati jednu od veličina. Brojne vrednosti u vidu razlomka korisnik unosi putem tastature. Nakon popunjavanja treba kliknuti na taster „UVRSTI“ za unos u izraz.

Y_A Y_B Y_C F q M Uneti: + $\frac{1}{2}$ $\frac{1}{2}$ UVRSTI L L²

(2) $\sum Y_i = 0$

(3) $\sum M_A = 0$

Slika 1. Rešenje interfejsa za unos rezultata u vidu matematičkog izraza

Važno je razumeti da korisnik/student treba da je u mogućnosti da sa lakoćom koristi softver bez dodatnog

predznanja o njegovom funkcionisanju kao i da jedino predznanje koje korisnik treba da poseduje je elementarni rad na računaru. Brzina unosa rešenja je takođe bitan koncept. Sporo i teško unošenje rezultata ili rešenja dovodi do frustracije kod korisnika. Primer za sporo unošenje rešenja je pisanje matematičkih izraza u programskim jezicima kao što su LaTeX. Takođe, za unošenje izraza ovom metodom potrebno je predznaje programskog jezika.

Slika 2. Konvencionalni način unošenja teksta

Rešenje koje napisano u ovom softveru sastoji se iz kombinacije konvencionalnih metoda unutar rešenja zadatka. Naime, rešenje zadatka je podeljeno u sekcije koje čine celinu. Svaka celina u sebi treba da sadrži jedan ili najviše dva koncepta. Nakon urađene celine (za koju ne treba više od nekoliko minuta) korisnik daje odgovor i nastavlja dalje. Odgovor se uglavnom sastoji iz ponuđenih odgovora ili „Drag and drop“ načina popunjavanja testa.

Slika 3. Primer unosa rezultata u vidu matematičkog izraza

"Real-time" renderovanje teksta unutar aplikacije znatno olakšava unos rezultata. Pod ovim pojmom podrazumeva se da se korisniku pokazuje tekst u pravom vremenu prateći njegov unos.

Nakon urađenog testa korisnik zatvara test sa „predajom“ nakon čega softver pokazuje rezultate i ocenjuje rad.

3. POVRATNA INFORMACIJA

Jedan od najznačajnijih nedostataka u korišćenju softverskih alata kod učenja i/ili testiranja je povratna informacija. Očekivano je da korisnik ne odgovori na svako pitanje tačno tokom procesa učenja. Kod konvencionalnog načina prenošenja znanja u ovakvim situacijama nastavnik je taj koji, na samo da ukazuje na netačnost odgovora, već obrazloži zašto je odgovor netačan i uputi na bolji način zamišljanja ili sam tačan odgovor. Ovo je veliki problem tokom pisanja softvera ali ne i nemoguć!

Slika 4. Primer povratne informacije

Upravo zbog podele testa na manje celine, moguće je primetiti u kom koraku korisnik pravi grešku. Takođe je moguće prepoznati šablon nakon nekoliko sličnih grešaka koje se često javljaju. Softver u ovim situacijama treba da pruži povratnu informaciju kako bi uputio korisnika ka tačnom odgovoru kao i obrazloženju istog [5].

Uneti: $\pm \frac{1}{1}$ $\frac{7}{2}$

(2) $\sum Y_i = 0$ $\frac{Y_A + Y_B + Y_C + F - F - qL}{0} = 0$ ✓

(3) $\sum M_A = 0$ $\frac{Y_B 2L + Y_C 4L + FL - F 3L - M - q \frac{7}{2} L^2}{0} = 0$ ✗

Pogrešno uneto rastojanje!

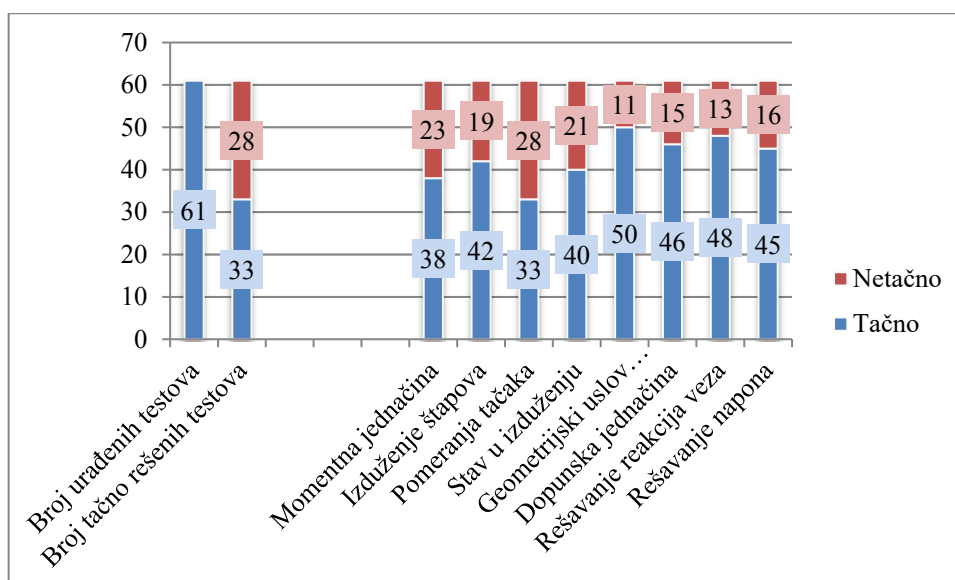
Pogrešno uneto rastojanje!

Slika 5. Primer povratne informacije

Jedna ovakva celina prikazana je na slici 5. Ona se sastoji od pisanja jednačina ravnoteže za zadati problem. Primer pokazuje povratnu informaciju softvera nakon netačno unešenih rastojanja u momentnoj jednačini.

4. AUTOMATIZOVANO SAMOTESTIRANJE

Pod pojmom samotestiranje podrazumeva se da student rešava test samostalno bez nadzora nastavnika ili mentora. Upravo ova vrsta testiranja je podstakla izradu sotverskog rešenja koji je naveden. Jedna od vodećih prednosti je mogućnost da student mnogo puta ponavlja proces samotestiranja i dobija povratnu informaciju o svom nivou znanja iz date oblasti. Ovaj metod dovodi do realnije slike o trenutnom nivou znanja studenta. Takođe, ovaj metod doprinosi stečenju znanja iz date oblasti bez učešća nastavnika.



Slika 6. Primer praćenja progresije

Nakon urađenih testova u dužem vremenskom periodu, student može samostalno da prati napredak u nivou znanja i veštine rešavanja zadataka (slika 6).

Rezultati dobijeni u [6] u kojima se porede konvencionalne metode testiranja i „e-testiranje“ pokazuju da je razlika u iskazivanju znanja neznatna, međutim motivacija studenata je dva puta veća kod elektronskih testova. Komunikacija sa nastavnikom je dvostruko manja a dostupnost rezultata testa je znatno veća tokom korišćenja elektronskih testova.

5. ZAKLJUČAK

Unapređenje procesa testiranja i učenja kroz softverske pakete daje studentu autonomiju tokom akumuliranja znanja tokom trajanja kursa. Povratna informacija iz softvera je najznačajniji pravac daljeg razvoja i istraživanja iz razloga što upravo ona čini da se testiranje, a ujedno i prenošenje znanja vrši bez prisustva nastavnika. Koncept automatizovanog testiranja upravo unapređuje pomenuti proces. Implementacija modernih načina učenja je, nažalost, spora i zahteva mnogo rada u samoj pripremi materijala.

6. LITERATURA

- [1] Brooks, David W., et al. "Automated Testing." *Journal of Science Education and Technology*, vol. 12, no. 2, 2003, pp. 183–86. *JSTOR*, <http://www.jstor.org/stable/40186605>. Accessed 7 Jan. 2023.
- [2] J. Skalka, M. Drlík and J. Obonya, "Automated Assessment in Learning and Teaching Programming Languages using Virtual Learning Environment," *2019 IEEE Global Engineering Education Conference (EDUCON)*, 2019, pp. 689-697, doi: 10.1109/EDUCON.2019.8725127.
- [3] Armin Berecki, Sanja Ožvat, *POVEZANOST MULTIMEDIJALNOG SADRŽAJA UNUTAR KURIKULUMA JEDNOG KURSA*, XXVIII Skup TRENDovi RAZVOJA: „UNIVERZITETSKO OBRAZOVANJE ZA PRIVREDU”, Kopaonik, 14 - 17. 02. 2022
- [4] Wengran Wang, Chenhao Zhang, Andreas Stahlbauer, Gordon Fraser, and Thomas Price. 2021. *SnapCheck: Automated Testing for Snap! Programs*. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA
- [5] Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2016. *Towards a Systematic Review of Automated Feedback Generation for Programming Exercises*. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16)*. Association for Computing Machinery, New York, NY, USA, 41–46.
- [6] Milica Tepšić , Tanja Borovnica, Slobodanka Bakić, *SISTEMI ZA ELEKTRONSKO TESTIRANJE ZNANJA STUDENATA*, COBISS.RS-ID 5446936, Ministarstvo uprave i lokalne samouprave Republike Srpske, 78000 Banja Luka, Bosna i Hercegovina