

JEZIK SPECIFIČAN ZA DOMEN MAKROA ZA TASTATURE**KEYBOARD MACRO DOMAIN SPECIFIC LANGUAGE**Dušan Janošević, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U radu je predstavljena implementacija jezika specifičnog za domen (DSL) makroa za tastature, kao i analiza domena makroa za tastature. Opisan je alat koji koristi ovaj domen i implementiran koristeći programski jezik Python i paket *textX* za parsiranje gramatike jezika specifičnog za domen, kao i za samo čitanje skripti napisanih datim jezikom specifičnim za domen. Glavni cilj projekta jeste napraviti jednostavni alat za pisanje makroa za tastature koji je intuitivan i zahteva niži nivo programerskog znanja.

Ključne reči: *DSL, textX, Python, alat, makro*

Abstract – The paper presents a domain-specific language (DSL) implementation of keyboard macros, as well as an analysis of the keyboard macros domain. A tool using this domain is described, implemented using the Python programming language and the *textX* package for parsing the grammar of a domain-specific language. The main goal of the project is to create a simple tool for writing macros for keyboards that is intuitive and requires a lower level of programming knowledge

Keywords: *DSL, textX, Python, tool, macro*

1. UVOD

Makroi za tastaturu predstavljaju unapred definisane nizove komandi koji omogućavaju automatsko izvršavanje više radnji na računaru jednim pokretom ili prečicom. Iako su svi makroi prečice, nisu sve prečice makroi, razlikuju se po složenosti. Prečice obično aktiviraju jednu funkciju dok su makroi kompleksniji i aktiviraju više radnji [3].

U savremenom računarstvu makroi su postali neizostavan deo svakodnevnog rada na računaru, naročito kod naprednijih korisnika koji žele da optimizuju svoj rad i povećaju svoju produktivnost. Pored makroa na tastaturi često se koriste i makroi vezani za miš [3].

Bitan su aspekt kod korisnika računara koji imaju ponavljajuće radnje a vremenom su se proširili i na video igre.

Uvođenjem kompleksnijih softverskih sistema raste i potreba za kompleksnijim makroima, posebno onima koje korisnici sami mogu definisati. Istraživanje sprovedeno na uzorku od 82 korisnika pokazalo je da postoji snažna veza između stepena svakodnevnog korišćenja računara i upotrebe makroa [3].

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Igor Dejanović, red. prof.

2. ISTORIJSKI RAZVOJ**2.1. Formatiranje teksta**

Prečice su nastale pre pojave računara, još na mehaničkim pisaćim mašinama. Prva značajna implementacija prečica bila je na Remingtonovoj pisaćoj mašini koja je koristila taster *Shift* za promenu veličine slova. U digitalnom obliku poznate su još i prečice za kopiranje, lepljenje i sečenje teksta [1].

Do 1980-ih prečice su postale standardna funkcionalnost. Uvedena je prečica *Ctrl+Alt+Del* 1981. Godine dok su *Alt+Tab* i *Undo* pojavili u kasnijim verzijama *Windows* operativnog Sistema

2.2. Makroi

Pojam makroa u računarskom smislu potiče iz 1960-ih kada su korišćeni u asemblerskim jezicima kao makro instrukcije koje smanjuju količinu izvorne programske logike. Tokom 1980-ih makroi se pojavljuju u alatima poput *SmartKey* i *ProKey*.

2.2.1. Period 1980-ih

Pojavljaju se prvi makroi u programima za tabelarne proračune, kao što je *lotus 1-2-3*. *Microsoft office* uvodi jednostavne makroe kroz alat *WordBasic*. Prvi program za snimanje pokreta miša i tastature zove se *AutoIt*.

2.2.2. Period 1990-ih

Microsoft pravi *VBA* alat za automatizaciju u *Office* programima. Makroi se počinju koristiti u video igrama gde jedno dugme može pokrenuti više radnji. Program *Macro Express* omogućava pravljenje makroa bez znanja programiranja.

2.2.3. Period 2000-ih

Alati kao što su *AutoHotKey* i *RPA* proširuju svoju primenu makroa na poslovne zadatke. Automatizacija polako prelazi iz licne upotrebe u poslovne tokove rada.

2.2.4. Period 2010-ih

Makroi se sele u *cloud* okruženja. *Selenium* omogućava automatizaciju *web* zadataka. Počinje integracija veštačke inteligencije za prepoznavanje elemenata i pokretanje makroa.

2.2.5 Period 2020-ih

Microsoft uvodi *Power Automate* u *Windows*. Makroi se široko primenjuju u industriji i uslugama, AI alati kao *ChatGPT* i *Copilot* pojednostavljaju kreiranje skripti. *VBA* i dalje ima snažnu ulogu a sigurnosni problemi su velikim delom rešeni [2, 4].

6.1 main.py

Glavni fajl projekta, sadrži *main()* funkciju od koje otpočinje izvršavanje. U njoj se prvo učitavaju sve makro skripte iz projektnog direktorijuma funkcijom *load_macros()*. Nakon čega se vrši provera cikličnosti pomoću klase *Checker*. Zatim se izabrana skripta šalje na izvršavanje kroz funkciju *interpret()*. Sam meta-model se učitava pre početka rada funkcija.

6.2. Definicija cli alata

Definiše drugi način pokretanja skripti preko komandne linije umesto pokretanja skripte kao Python projekta. Ulazna tačka je *cli.py* odakle se poziva definicija skripte iz fajla *type_macro.py*, koja učitava meta-model i definiše funkcije za učitavanje i izvršavanje makro skripte. Korisnik samo navodi putanju do željene skripte kao argument komandnog alata.

6.3. Checker klasa za proveru cikličnosti

Veoma bitna klasa u projektu. Služi za proveru cikličnosti poziva makroa, odnosno da li možemo ući u beskonačnu petlju poziva makroa. Ova pojava je potencijalno opasna jer može dovesti do gubitka kontrole nad računarom. Zato je u sam alat ugrađen bezbednosni makro za zaustavljanje svih akcija (*Alt+F12*).

Glavna funkcija je *detect_cycle()* koja primenjuje algoritam dubinske pretrage (DFS) nad stablom poziva. Za svaki makro proverava se da li aktivira drugi makro, i ako je tako taj odnos se prati rekurzivno. Ako tokom pretrage nađemo na makro koji je već bio u putanji pretrage, otkrivena je cikličnost i algoritam vraća upozorenje.

Ako nijedan put ne odvede do takve situacije, makro se beleži kao bezbedan i isključuje se iz narednih provera. Ovakva provera obezbeđuje stabilnost sistema i sprečava slučajne beskonačne petlje u automatizaciji.

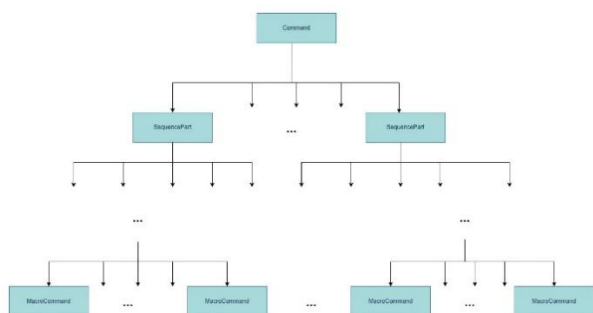
6.4. Python klase modela

Selekcija u kojoj se posebno opisuju sve klase domena u Python obliku, opisuju se njihovi atributi kao i funkcije koje poseduju date klase.

Nakon što se sam model učitava preko klase *MacroGroup* koja suštinski poseduje celokupnu skriptu sa makroima, celokupna stabla koja predstavljaju makroe se svode na klasu *FullMacro* koja strukturu stabla svo ina listu akcija.

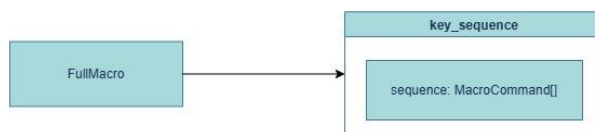
6.5. Raspeljavanje

Raspeljavanje je proces svodenja stabla akcija jednog makroa na listu akcija. Na početku modela, svaki pojedinačni makro poseduje stablo akcija koje obuhvata izvršavanje makroa. Ovaj oblik nije baš pogodan za izvršavanje i može se raspeljati zbog jednostavnosti. Na slici 2. prikazan je primer jednog stabla makroa.



Slika 2. Primer stabla akcija jednog makroa

Nakon što se izvrši raspeljavanje, dobiće se instanca klase *FullMacro* koja će u sebi sadržati raspeljavanu listu akcija koje su se nalazile u stablu (Slika 3.).



Slika 3. Primer raspeljane sekvence akcija preko klase *FullMacro*

6.6. Interpreter

Sve počinje od funkcije *interpret()* koja pokreće slušanje sa tastature ili slušanje specifičnog piksela na ekranu u zavisnosti od tipa makroa, svako slušanje makroa poseduje svoju zasebnu nit. Ako je makro zasnovan na tasterima, koristi se funkcija *listen_to_macro()*, a ako reaguje na boju piksela poziva se funkcija *listen_to_pixel()*, niti koje su zasnovane ovim funkcijama zovemo nitima slušačima.

Dok se slušaju događaji, odvojena nit iz paketa *pynput* prati kada je taster pritisnut ili otpušten i u zavisnosti od toga puno odnosno prazni set pritisnutih tastera. Kada se ispunе uslovi za pokretanje makroa, nit slušač će pokrenuti novu nit koja će izvršavati pritiske tastera definisane makroom koji je aktiviran preko funkcije *execute_macro_sequence()*.

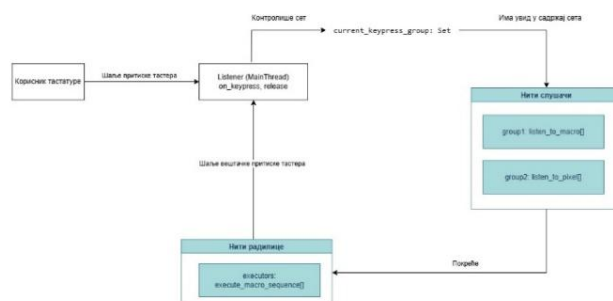
6.6.1. Multi-threading pojašnjeno

Do sada su spomenute niti koje su prisutne kod interpretera, ovde su definisani odnosu između njih kao i detaljniji opis tih niti.

Glavna nit prati pritiske tastera, ova nit potiče iz paketa *pynput*. Služi da ubacuje odnosno izbacuje tastere iz seta pritisnutih tastera.

Niti slušači su niti koje se pokreću u isto vreme kada i glavna nit. Ove niti prate sadržaj seta pritisnutih tastera i proveravaju da li je ispunjen uslov za pokretanje makro sekvence akcija.

Niti radilice su niti koje se instanciraju u nitima slušačima. Kada je uslov za aktivaciju makroa ispunjen, instancira se nova nit radilica koja će izvršavati akcije datog makroa, za to vreme, nit slušač nastavlja da sluša. Ovo znači da jedna nit slušač može da instancira više istih niti radilica.



Slika 4. Primer modela funkcionisanja niti interpretera

7. PRIMERI SKRIPTI DEFINISANI JEZIKOM

U ovom poglavlju opisani su konkretni primeri korišćenja skripti napisanih za korišćenje uz pomoć alata opisanog u ovom radu.

7.1. Otvaranje programa i puštanje muzike

Ova skripta sadrži više funkcija koje pomažu pri apstrakciji određenih sekvenci akcija i pomažu pri jednostavnije pisanju. Skripta poseduje makroe za otvaranje *Web* pretraživača Opera, makro za otvaranje programa za video igre *Steam* [14] kao i dve varijante prečica za puštanje muzike, jedna varijanta preko browsera, druga preko aplikacije za muziku *Spotify*.

```
keyPressDelay = 200;

loopTab(count: number): Loop(count, [tab]);

openOpera(): o,p,u,r,e,enter;
openSteam(): s,t,e,a,m,enter;
openSpotify(): s,p,o,t,i,f,y,enter;

typeYoutube(): y,u,t,u,enter;
playLoFi(): l,o,space,f,l,enter;

playLoFiYoutube(): openOpera(),sleep(200),typeYoutube(),sleep(200),loopTab(4),typeLoFi(),sleep(200),loopTab(23),enter;
playLoFiSpotify(): openSpotify(),sleep(200),ctrl+k,sleep(200),typeLoFi(),loopTab(2),enter,sleep(200),loopTab(2),enter;
playLoFi(isSpotify: boolean): {if(isSpotify): [playLoFiSpotify()]; else: [playLoFiYoutube()]};
```

Slika 5. Pomoćne funkcije za skriptu otvaranja programa i pokretanja muzike

```
Main {
  alt + o :sleep(300), command, sleep(100), openOpera();
  alt + s :sleep(300), command, sleep(100), openSteam();
  alt + l :sleep(300), command, sleep(100), playLoFi(false);
  alt + k :sleep(300), command, sleep(100), playLoFi(true);
};
```

Slika 6. Makroi za pokretanje muzike i specifičnih programa uz pozivanje pomoćnih funkcija

7.2. Skripta za video igru *Path of Exile*

Pokazuje pravu moć makroa z video igrama, sadrži dva makroa koja slušaju specifične pozicije piksela na ekranu. Ukoliko količina životnih poena ili količina magije heroja u igri padne ispod određenog nivoa, odnosno promeni se boja piksela bara, aktiviraće se specifičan makroa da taj bar napuni.

Pored toga sadrži i jednostavan makro za aktivaciju više magija preko prečice.

```
heal(): 1;
mana(): 5;

Main {
  onColor(#3e2627, 750, 1350,150) : heal();
  onColor(#16161d, 2690, 1350,150) : mana();
  alt + d : ctrl + q, ctrl + w, ctrl + e, ctrl + r;
};
```

Slika 7. Primer skripte za slušanje barova za igru *Path of Exile*[16] za aktivaciju moći

8. ZAKLJUČAK

Makroi predstavljaju unapred definisane sekvence akcija komandi koje značajno ubrzavaju rad na računaru. Njihov razvoj je prošao dug put. Od mehaničkih prečica do modernih alata koji mogu koristiti veštačku inteligenciju. Glavna prednost makroa je ušteda vremena, pritiskom nekoliko tastera mogu se pokrenuti složene radnje, što je posebno korisno u profesionalnom okruženju.

U radu je prikazano kako su alati za makroe evoluirali i kako se danas koriste u alatima poput *AutoHotKey*-a i *Macro Express*-a. Cilj projekta bio je da se napravi jednostavan, ali moćan alat prilagođen početnicima ali dovoljno fleksibilan za napredne korisnike. Fokus je stavljen na lakoću upotrebe, mada to ograničava rad na složenijim skriptama.

Projekat takođe uključuje proveru postojanja rekurzije među makroima kao i *multi-threading* mehanizam za paralelno izvršavanje, što omogućava bolje performanse.

Primeri upotrebe pokazuju primenu u svakodnevnom radu i igrama.

Unapređenja ovog alata obuhvataju proširenje sa grafičkim interfejsom, kontrolu miša, kao i prebacivanje alata na neki drugi programski jezik zbog performansi. Takođe interesantno proširenje bila bi podrška za više računara, odnosno kontrola više računara sa jednog glavnog.

9. LITERATURA

[1] <https://www.taskade.com/blog/history-keyboard-shortcuts-productivity>, pristup 26. Фебруар 2025.

[2]

[https://en.wikipedia.org/wiki/Macro_\(computer_science\)](https://en.wikipedia.org/wiki/Macro_(computer_science))

[3] Peres, S. C., Tamborello, F. P., Fleetwood, M. D., Chung, P., & Paige-Smith, D. L. (2004). *Keyboard Shortcut Usage: The Roles of Social Factors and Computer Experience. Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 48(5), 803–807. <https://journals.sagepub.com/doi/abs/10.1177/154193120404800513>

[4] <https://www.digit.in/features/general/digit-mag-the-origin-of-software-macros-53206.html>, pristup 1. Март 2025.

[5] <https://www.macrorecorder.com>, pristup 1. Март 2025.

[6] <https://www.macros.com>, pristup 1. Март 2025.

[7] <https://www.autohotkey.com>, pristup 1. Март 2025.

[8] <https://www.macrocreator.com>, pristup 8. Март 2025.

[9] <https://learn.microsoft.com/en-us/power-automate>, pristup 8. Март 2025.

[10] <https://www.Python.org/about/>, pristup 18. Април 2025.

[11] <https://textx.github.io/textX/>, pristup 18. Април 2025.

[12] <https://pynput.readthedocs.io/en/latest/>, pristup 17. Мај 2025.

[13] <https://pyautogui.readthedocs.io/en/latest/>, pristup 18. Мај 2025.

[14] <https://steam.fandom.com/wiki/Steam>, pristup 29. Јун 2025.

[15] <https://en.wikipedia.org/wiki/Spotify>, pristup 29. Јун 2025.

[16] <https://www.pathofexile.com>, pristup 29. Јун 2025.

Kratka biografija:



Dušan Janošević rođen je u Majdanpeku 2001. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo odbranio 2025. godine.

Kontakt: dušan.janosevic123@gmail.com