

Sistem za preporuku projektnih obrazaca na srpskom jeziku kao alat u softverskom inženjerstvu

Aleksandar Cenić/Dejan Rančić
Departman za Računarsku nauku
Elektronski fakultet
Niš, Srbija
acenic@elfak.rs/dejan.rancic@elfak.ni.ac.rs

Sažetak— Softversko inženjerstvo, kao disciplina ima za zadatak rešavanje određenih problema u samoj izradi, odnosno, projektovanju softvera. Ovaj rad se pretežno bavi primenom metoda mašinskog učenja kao osnovu sistema za preporuku projektnih obrazaca. Sistem preporuku bazira na zadatom projektnom zahtevu na srpskom jeziku. Pored preporuke sistem generiše grafički UML dijagram predloženog obrasca, takođe i okvire softverske dokumentacije. U ovome radu je, u prvom delu, napravljen pregled svih prethodnih metoda sistema za preporuku projektnih obrazaca, u drugom delu data je analiza kreiranog skupa podataka, u trećem delu opisana je izrada sistema, a na kraju su dati uporedni rezultati tri klasifikatora: K najbližih suseda, stablo odlučivanja i Multinomialni Bajesov klasifikator u dve metode.

Ključne riječi- *Softversko inženjerstvo - 1; Sistem za preporuke 2; projektni obrazac 3; Srpski jezik*

I. UVOD

Softversko inženjerstvo predstavlja inženjersku disciplinu koja obuhvata projektovanje, implementaciju i održavanje softvera. Uočavanje i primena nekoliko rešenja iznova i iznova na različite projektne probleme, dovodi nas do pojma projektnih obrazaca. Projektni obrasci predstavljaju šablone čijom primenom se mogu rešavati problemi u projektovanju softvera. U svrhu rešavanja problema korektnog izbora obrasca na projektni problem, u ovom radu ćemo predstaviti jedan tip sistema za preporuku projektnih obrazaca [1]. Pored sistema preporuke projektnih obrazaca, predloženi alat će generisati UML dijagrame klasa projektnih obrazca i templejte dokumenta, pre svega, dokumenta softverske specifikacije i arhitekturnog projekta. Rani sistem za preporuku projektnih obrazaca, bazirao se na poređenje opisa projektnog problema sa opisom projektnog obrasca iz GoF(Gang of Four) tabele [2]. Jos jedna metoda kod sistema za preporuku projektnih obrazaca jeste sistem baziran na slučaju [3], gde se na svaki problem dodaje i oznaka upotrebljenog obrasca što bi na neki način predstavljalo i skup podataka koji se može upotrebiti u mašinskom ili dubokom učenju. Još jedna metoda jeste, sistem zasnovan na pitanjima [3] gde se kombinuje metoda zasnovana na slučaju i interakcije ljudi tj. softverskog inženjera. Jedan od radova opisuje alat za generisanje UML dijagrama uz pomoć NLP(Natural language processing) metoda i heurističkih pravila na osnovu softverskih zahteva na prirodnom jeziku [4], takođe jedan srodan rad u kome se ispituje sposobnost GPT modela za generisanje sekvencijalnog dijagrama [5]. U jednom od radova, za sistem preporuke projektnih obrazaca koriste se

tehnike mašinskog učenja [6]. U ovome radu biće, u prvom delu, data definicija sistema za preporuke, opis dosada korišćenih pristupa kod sistema preporuke projektnih obrazaca i biće pomenuti softverski alati koji postoje kao asistencija u softverskom projektovanju. U drugom delu, biće opisan pristup baziran na mašinskom učenju koji smo koristili u izradi samog alata i način generisanja skupa podataka. U trećem delu rada biće prikazani rezultati postignuti implementiranim alatom tj. sistemom. Konačno, u četvrtom i poslednjem delu rada biće dat zaključak i smer budućeg rada.

II. SISTEMI PREPORUKE

Sistemi preporuke, u suštini, predstavljaju jedan način rangiranja i selekcije, odnosno filtriranja određenih objekata, pristupa ili metoda [7]. Sam sistem preporuke projektnih obrazaca predstavlja alat iz domena vestačke inteligencije.

A. Definicija sistema za preporuku projektnih obrazaca

Za dati ulaz P; problem opisan prirodnim jezikom, sistem S treba dati određeni izlaz I. Iz navedene definicije, vidimo da nam je ulazni podatak u sistemu projektni problem opisan prirodnim jezikom. Tako da se za implementaciju sistema koriste NLP metode. U ovom delu, opisaćemo neke od postojećih metoda korišćenih za implementaciju sistema za preporuku projektnih obrazaca.

B. Starije metode sistema za preporuku projektnih obrazaca

Starije metode kod sistema za preporuku projektnih obrazaca su pretežno dvofazne metode koje se baziraju ili na kodicionalnom stablu [8], ili na matricama sličnosti koje sadrže ključne termine iz opisa obrazaca [9], a skoro svaka od tih metoda interaguje sa softverskim arhitektom. Još jedna od metoda koje je dobro pomenuti jeste metoda klasterizacije. Kod ovog metoda se najpre projektni obrazac kategorizuje po tipu tj. da li je kreacioni strukturni ili obrazac ponašanja. U tu svrhu, koristio se Fuzzy c-means algoritam koji predstavlja algoritam nenadgledanog učenja [10]. Nakon kategorizacije projektnog obrasca po tipu, računa se kosinusna sličnost između projektnog problema i opisa projektnog obrasca. Takođe, može se pomenuti i metod baziran na pitanjima i odgovorima, koji na osnovu projektnog problema, generičkih pitanja postavljenih softverskom arhitekti donosi odluku o

izboru obrasca, tako da postoji malo šira klasifikacija sistema za preporuku [11].

C. Semantički metod kod sistema za preporuku projektnih obrazaca

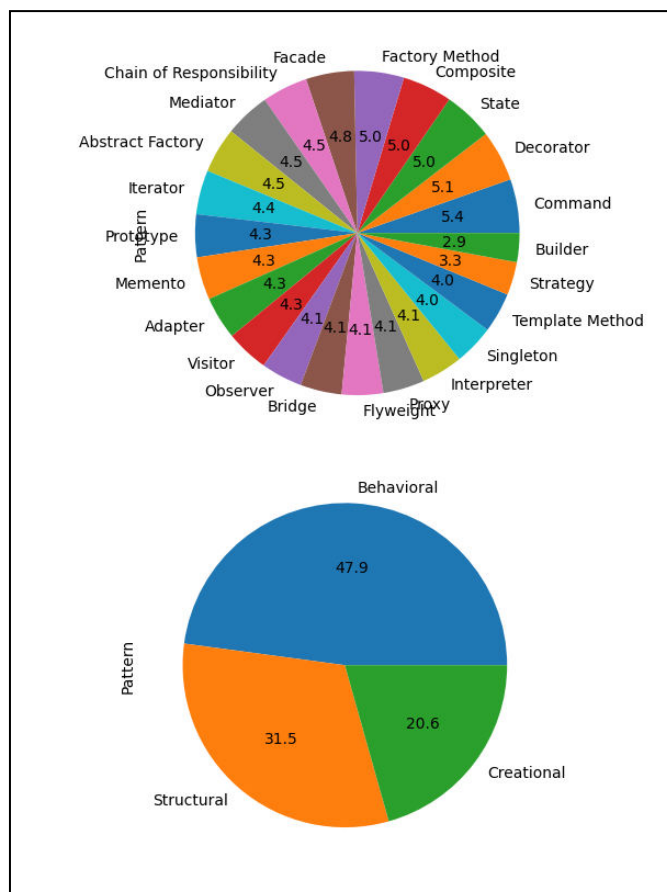
Jedan od relativno novih metoda jeste semantički pristup kod sistema za preporuku obrazaca. Kod ovog pristupa gde se reči iz opisa obrazaca pretvaraju embedding vektore kako bi se bolje uočila veza odnosno proračunala sličnost između zahteva i opisa obrasca [12]. Naravno, da danas u doba velikih jezičkih modela ovo bi se moglo smatrati i zastarelom metodom.

D. Metod baziran na velikim jezičkim modelima

Sa samom pojavom velikih jezičkih modela sve oblasti veštačke inteligencije su dobile veliku tačnost u izvršenju zadataka, pa se naravno upotrebljavaju i kod sistema preporuke projektnih obrazaca kao nova generacija sistema preporuke [13]-[14].

III. PREDLOŽENI METOD I IMPLEMENTACIJA

Radi implementacije našeg sistema za preporuku projektnih obrazaca na srpskom jeziku u sistem preporuke bice integrisan u softver za crtanje UML klasa, generisanje koda, prpratnte dokumentacije i zajedno uz softver sačinjavaće jedan alat koji predstavlja dobru asistenciju pri softverskom projektovanju. Sam softver za generisanje je takođe izrađen kao deo celokupnog alata i izrađen je Qt framework-u. Celokupan alat ima dve zasebne celine, softver za iscertavanje UML dijagrama i obrazaca, dok drugi deo sačinjava sam sistem preporuke napravljen u python-u. Kod sistema preporuke koristicemo metode nadgledanog učenja, odnosno koristicemo klasifikatore, preciznije, upotrebićemo tri klasifikatora, na dva različita načina i uporediti rezultate. Kako ćemo koristiti nadgledano učenje biće nam potreban skup podataka. Sam sistem preporuke koji ćemo implementirati zasniva se na srpskom jeziku, a kako za srpski jezik postoji mali broj skupova podataka, pogotovo za ovu vrstu problema, moraćemo kreirati skup podataka nad kojim ćemo trenirati klasifikatore. Skup podataka kreiran je prikupljanjem zadataka sa blanketa ispita objektivnog orijentisanog projektovanja, naravno jednim delom, jer je bilo malo podataka za solidan skup podataka. Drugim i većim delom kreiran je uz pomoć velikog jezičkog modela GPT 3.5. Kreirani skup podataka podeljen je u dva dela, prvi deo skupa jeste skup na kome su zahtevi obeleženi po tipu projektnog obrasca, odnosno skup koji sadrži labela da li projektni obrazac za dati zahtev pripada tipu kreacionog, strukturnog, ili obrascu ponašanja. Drugi deo skupa jeste direktan gde je za svaki projektni zahtev data labela određenog obrasca. Upravo zbog upotrebe dva različita načina rada predloženog sistema, podelili smo kreirani skup u dva dela. Oba dela imaju iste podatke, tj. projektne zahteve, dok se razlikuju samo u labelama. U kreiranom skupu podataka postoje 23 različita tipa labela, odnosno, sistem preporuke će preporučivati samo 23 osnovna projektna obrasca iz GoF tabele. Razlog za podelu skupa u dva dela, jeste i taj što se kod klasifikacije na skupovima podataka koji imaju veliki broj klasa dobija mala tačnost.



Slika 1. Raspodela skupa podataka. (

Sl.1 nam pokazuje da je prvi skup većim delom balansiran, a da primeri za obrasce Builder i Strategy su najmanje zastupljeni i još nam pokazuje da skup obeležen po tipu obrasca nije balansiran, što je i normalno jer veći deo obrazaca iz GoF tabele su obrasci ponašanja. Pri kreiranju sistema za preporuku projektnog obrasca isprobaćemo dva pristupa i uporediti dobijene rezultate. Prvi pristup koji ćemo upotrebiti baziran je na metodi ranije pomenutoj u ovom radu, a to je metoda klasifikacije i kategorizacije teksta u kojoj su se zahtevi kategorizovali po tipu projektnog obrasca uz pomoć Fuzzy c-means algoritma, a nakon toga se računa kosinusna sličnost između projektnog zahteva i opisa projektnog obrasca iz GoF tabele. Kod predloženog metoda koristimo metode za klasifikaciju, tj. klasifikatore za određivanje tipa obrasca za projektni zahtev, a nakon toga ponovo koristimo klasifikatore trenirane na podskupu podataka za određeni tip obrazaca, odnosno za tip dobijen iz prethodnog koraka. Kako se projektni zahtevi sistemu zadaju na prirodnom jeziku, a i sam skup podataka jeste NLP skup podataka, za predprocesiranje podataka i izvlačenje obeležja koristimo NLP metode za procesiranje.

Za potrebe ovog sistema koristićemo sledeće metode:

Tokenizaciju,

Uklanjanje stop reči,

Računanje TF-IDF mera kao obeležja skupa.

Prva i osnovna metoda kod procesiranja teksta jeste tokenizacija teksta, gde se dati dokument ili sentenca lomi, odnosno transformiše u niz reči. Nakon tokenizacije skupa iz dobijenog niza reči uklanjaju se najfrekventnije reči koje ne nose nikakve relevantne informacije. Za zadatak tokenizacije upotrebljen je ReldiTokeniser, tokenizer za SLOVENSKE jezike u našem slučaju srpskog jezika [15]. Poslednja faza kod preprocesiranja jeste računanje TF-IDF mere. TF-IDF mera. TF-IDF mera predstavlja metodu ekstrakcije obeležja tekstualnih podataka koja se bazira na frekvenci termina [10]. Skup podataka pri treniranju klasifikatora, podeljen je `train_test_split` metodom iz `sklearn` biblioteke samo na trening i test podskupe zbog malog obima podataka i to 60% skupa za treniranje, a 40% za testiranje. Kod predprocesiranja nismo koristili stemovanje jer bi smanjilo vokabular, koji je veličine 2200 reči. Izbor parametara je izvršen manualnom selekcijom. Za broj suseda uzeto je 10, a za dubinu stabla 7.

IV. EVALUACIJA REZULTATA

U ovom poglavlju ćemo dati pregled dobijenih rezultata u oba predložena metoda.

A. Metoda selekcije obrasca po tipu

TABELA I. REZULTATI KLASIFIKACIJE ZAHTEVA PO TIPU PROJEKTHNIH OBRASCA

<i>Klasifikator</i>	<i>Tačnost</i>
klasifikator K najbližih suseda	88%
Stablo odlučivanja	71%
Multinomialni Bajesov klasifikator	87%

Vidimo da je klasifikator K najbližih suseda dao najveću tačnost od 88%.

TABELA II. REZULTATI KLASIFIKACIJE ZAHTEVA KREACIONIH PROJEKTHNIH OBRASCA

<i>Klasifikator</i>	<i>Tačnost</i>
klasifikator K najbližih suseda	90%
Stablo odlučivanja	85%
Multinomialni Bajesov klasifikator	75%

Vidimo da je i u slučaju kreacionih obrazaca klasifikator K najbližih suseda dao najveću tačnost od 90%.

TABELA III. REZULTATI KLASIFIKACIJE ZAHTEVA STRUKTURNIH PROJEKTHNIH OBRASCA

<i>Klasifikator</i>	<i>Tačnost</i>
klasifikator K najbližih suseda	86%
Stablo odlučivanja	79%
Multinomialni Bajesov klasifikator	83%

Vidimo da je i u slučaju strukturnih obrazaca klasifikator K najbližih suseda dao najveću tačnost od 86%.

TABELA IV. REZULTATI KLASIFIKACIJE ZAHTEVA OBRASCA PONAŠANJA

<i>Klasifikator</i>	<i>Tačnost</i>
klasifikator K najbližih suseda	81%
Stablo odlučivanja	66%
Multinomialni Bajesov klasifikator	83%

Vidimo da se u slučaju obrazaca situacija promenila i da je Multinomialni Bajesov klasifikator dao najveću tačnost od 83%. Kod metode selekcije obrazaca po tipu, odnosno klasifikacije tipa obrasca po projektnom zahtevu, klasifikacije kreacionih obrazaca i klasifikacije strukturnih obrazaca, najbolje rezultate postigao je klasifikator K najbližih suseda, jedino u slučaju obrazaca ponašanja, najbolje rezultate postigao je Multinomialni Bajesov klasifikator. U narednom delu daćemo pregled rezultata direktne metode selekcije projektnih obrazaca.

B. Direktna metoda selekcije obrasca

TABELA V. REZULTATI KLASIFIKACIJE ZAHTEVA OBRAZACA PONAŠANJA

Klasifikator	Tačnost
klasifikator K najbližih suseda	78%
Stablo odlučivanja	29%
Multinomialni Bajesov klasifikator	69%

Čak i kod direktnog pristupa istakao se klasifikator K najbližih suseda i dao najveću tačnost od 79%. Nakon analize dobijenih rezultata možemo reći da je skoro u oba metoda najveću tačnost postigao klasifikator K najbližih suseda i da je metoda selekcije po tipu dala bolje rezultate, kako se i očekivalo, mada ne možemo zanemariti činjenicu da je sve kod metode selekcije po tipu zavisno upravo od klasifikacije projektnog zahteva po tipu obrasca.

V. ZAKLJUČAK

Sistemi preporuke predstavljaju jako korisne alate u izboru određenih artikala, alata, pa i metoda u slučaju projektovanja. U ovom radu obrađeni su sistemi preporuke projektnih obrazaca kao alati u softverskom inženjerstvu. Izrađeni sistem u ovom radu sačinjen je od dve komponente sistema preporuke, i aplikacije za crtanje UML dijagrama klasa. Zajedno, ove dve komponente sačinjavaju jedan dobar alat za oblast softverskog projektovanja, jer na osnovu specifikacije zahteva, alat predlaže projektni obrazac, i na osnovu svega toga generiše i projektnu dokumentaciju, specifikaciju zahteva i arhitekturni projekat. Za izradu sistema preporuke korišćeno je nadgledano učenje, odnosno tri različita klasifikatora. Korišćeni su klasifikator K najbližih suseda, stablo odlučivanja i Multinomialni Bajesov klasifikator. Korišćena su dva pristupa, pristup selekcije po tipu i pristup direktne selekcije. Kod oba pristupa najveću tačnost dao je klasifikator K najbližih suseda, osim u slučaju selekcije po tipu za obrasce ponašanja gde je najveću tačnost postigao Multinomialni Bajesov klasifikator.

LITERATURA

- [1] F. Palma, H. Farzin, Y. -G. Guéhéneuc and N. Moha, "Recommendation system for design patterns in software development: An DPR overview," 2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE), Zurich, Switzerland, 2012, pp. 1-5.
- [2] S. Hussain, J. Keung and A. A. Khan, "The Effect of Gang-of-Four Design Patterns Usage on Design Quality Attributes," 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), Prague, Czech Republic, 2017, pp. 263-273.
- [3] U. Celikkan and D. Bozoklar, "A Consolidated Approach for Design Pattern Recommendation," 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 2019, pp. 1-6.

- [4] E. A. Abdelnabi, A. M. Maatuk, T. M. Abdelaziz and S. M. Elakeili, "Generating UML Class Diagram using NLP Techniques and Heuristic Rules," 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, 2020, pp. 277-282.
- [5] A. Ferrari, S. Abualhaija, C. Arora, "Model Generation with LLMs: From Requirements to UML Sequence Diagrams", IEEE 32nd International Requirements Engineering Conference Workshops, 2024, pp. 291-300.
- [6] E. Dias Canedo, B. Cordeiro Mendes, "Software Requirements Classification Using Machine Learning Algorithms", 2020, Entropy.
- [7] A. Braham, M. Khemaja, F. Buendia, F. Gargouri, "A Hybrid Recommender System for HCI DesignPattern Recommendation", 2021, Applied Sciences.
- [8] Y. Clara K. Youssef, Farida M. Ahmed, Hashem M. Hashem, Veronia E. Talaat, Nada Shorim, "GQM-based Tree Model for Automatic Recommendation of Design Pattern Category," Association for Computing Machinery, Association for Computing Machinery, 2020, pp. 11-13.
- [9] I. Issaoui, N. Bouassida, H. Ben-Abdallah, "A New Approach for Interactive Design Pattern Recommendation", Lecture Notes on Software Engineering, 2015, Vol. 3, No. 3.
- [10] S. Hussain, J. Keung, A. Ali Khan, Software design patterns classification and selection using text categorization approach, Applied Soft Computing Journal, 2017, pp. 225-244.
- [11] A. Naghdipour, S. M. Hossien Hasheminejad and M. Reza Keyvanpour, "DPSA: A Brief Review for Design Pattern Selection Approaches," 2021 26th International Computer Conference, Computer Society of Iran (CSICC), Tehran, Iran, 2021, pp. 1-6.
- [12] D. Liu, H. Jiang, X. Li, Z. Ren, L. Qiao and Z. Ding, "DPWord2Vec: Better Representation of Design Patterns in Semantics," in IEEE Transactions on Software Engineering, April 2022, vol. 48, no. 4, pp. 1228-1248.
- [13] Z. Zhao et al., "Recommender Systems in the Era of Large Language Models (LLMs)," in IEEE Transactions on Knowledge and Data Engineering, Nov. 2024, vol. 36, no. 11, pp. 6889-6907.
- [14] Wang, Qi et al. "Towards Next-Generation LLM-based Recommender Systems: A Survey and Beyond." ArXiv abs/2410.19744 (2024): n. pag.
- [15] V. Batanović, N. Ljubešić, T. Samardžić, M. Miličević Petrović, "Otvoreni resursi i tehnologije za obradu srpskog jezika", Conference: Primena slobodnog softvera i otvorenog hardvera, 2020.

ABSTRACT

Software engineering, as a discipline, has the task of solving certain problems in the production itself, that is, designing software. This paper mainly deals with the application of machine learning methods as the basis of a system for recommending project patterns. Recommendation system based on the given project request in the Serbian language. In addition to the recommendation, the system generates a graphic UML diagram of the proposed form, as well as software documentation frameworks. In this paper, in the second part, an overview of all previous methods of the system for recommending project forms was made, in the third part, an analysis of the created data set was given, and the method used was described. and at the end the comparative results of three classifiers are given: knn, decision tree and Multinomial Bayesian classifier in two methods. In both methods, the highest accuracy was given by the knn classifier

A SYSTEM FOR RECOMMENDING PROJECT FORMS IN THE SERBIAN LANGUAGE AS A TOOL IN SOFTWARE ENGINEERING

Aleksandar Cenić, Dejan Rančić