

GENERATOR KODA ZA MOBILNE APLIKACIJE CODE GENERATOR FOR MOBILE APPLICATIONS

Olivera Mirilović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu je predstavljen softverski alat vođen modelima za specifikaciju i generisanje koda mobilnih aplikacija. Rad obuhvata analizu postojećih rešenja koja koriste veštačku inteligenciju, low-code platforme i cross-platform radne okvire. Razvijeni meta-model i generator koda omogućavaju generisanje mobilnih aplikacija. Cilj ovog rada je da pokaže da implementirani generator može, do određene mere, omogućiti trenutnim ili budućim programerima brži razvoj mobilnih aplikacija, kao i da doprinese proširenju njihovog znanja i iskustva.

Ključne reči: Inženjerstvo vođeno modelima, MDE, meta-model, MagicDraw, generator, mobilne aplikacije

Abstract – This paper presents a model-driven software tool for the specification and code generation of mobile applications. It includes an analysis of existing solutions that utilize artificial intelligence, low-code platforms, and cross-platform frameworks. The developed meta-model and code generator enable the generation of mobile applications. The goal of this paper is to demonstrate that the implemented generator can, to a certain extent, facilitate the development of mobile applications for current and future developers, as well as contribute to the expansion of their knowledge and experience.

Keywords: Model driven engineering, MDE, meta-model, MagicDraw, generators, mobile applications

1. UVOD

Generatori mobilnih aplikacija postali su sve istaknutiji poslednjih godina, nudeći pristupačna sredstva za preduzeća, pojedince i netehničke korisnike da kreiraju aplikacije bez potrebe za opsežnim znanjem kodiranja. Ove platforme su pojednostavile razvoj aplikacija obezbeđujući unapred izgrađene šablone, lako dostupne interfejsa i prilagodive funkcije. Sve veća upotreba mobilnih uređaja i potreba za efikasnim rešenjima koja će zadovoljiti zahteve korisnika dodatno su podstakli rast ovih platformi. Značaj generatora mobilnih aplikacija leži u njihovoj sposobnosti da ubrzaju proces razvoja.

Predstavljeni softverski alat sastoji se od sledećih modula.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Gordana Milosavljević, red. prof.

Prvi modul sadrži metamodel na osnovu kojeg se specificira model mobilne aplikacije, dok drugi modul, generator, obuhvata kolekciju šablona za generisanje izvršivog programskog koda na osnovu dobijenog modela iz prvog modula. Osnovni motiv za razvoj generatora mobilnih aplikacija je ubrzanje procesa njihovog razvoja, kao i podsticanje širem krugu korisnika da kreiraju i održavaju mobilne aplikacije.

2. POSTOJEĆA REŠENJA

Neki od pristupa koji se koriste pri razvoju generatora mobilnih aplikacija uključuju cross-platform radne okvire, low-code platforme i veštačku inteligenciju. Među ovim pristupima, razvoj softvera vođen modelom privukao je pažnju zbog svog potencijala da automatizuje značajan deo procesa razvoja mobilnih aplikacija.

Jedna od istaknutih metoda je upotreba cross-platform radnih okvira koji omogućavaju programerima da pišu aplikacije na jednom jeziku koji se zatim može kompajlirati u kod specifičan za platformu. Radni okvir u ovoj kategoriji vredan pomena je Apache Cordova [1], koji omogućava razvoj mobilnih aplikacija koristeći HTML, CSS i JavaScript, a zatim ih umotava u izvorni kontejner. Prema ovom radu [2], ovaj pristup je značajno smanjio vreme i troškove razvoja mobilnih aplikacija omogućavajući da jedna baza koda funkcioniše na više platformi. I pored njegovih prednosti, postoje приметni problemi sa performansama, posebno u grafički intenzivnim aplikacijama koje mogu biti veoma ograničene.

Drugi pristup fokusira se na low-code platforme koje pružaju programerima mogućnost da kreiraju mobilne aplikacije putem grafičkih interfejsa i unapred izgrađenih komponenti. OutSystems [3] je dobar primer ovog pristupa, a njegove prednosti su istražene u radu [4]. Dok ove platforme čine razvoj pristupačnijim, one mogu ograničiti prilagodivost, jedinstvenost komponenti i mogu imati poteškoća da ispune zahteve performansi složenih aplikacija.

Napredak u okvirima veštačke inteligencije orijentisanim razvojnim okruženjima baziranim na modelima nedavno je došao u prvi plan razvoja mobilnih aplikacija. Istraživanje u radu [5] proučava kako metode vođene modelom, proširene veštačkom inteligencijom, mogu dalje optimizovati generisanje aplikacija sugerišući dizajn interfejsa i poboljšavajući efikasnost koda na osnovu unapred definisanih modela. Ova kombinacija MDE i veštačke inteligencije predstavlja oblast u razvoju koja obećava dalja smanjenja ručne intervencije u procesu kreiranja aplikacija.

3. KORIŠĆENE TEHNIKE I TEHNOLOGIJE

MagicDraw alat je korišćen za dobijanje podataka o elementima modela, a Java aplikacija, sa FreeMarker obrađivačem šablona, za generisanje same aplikacije. Rezultirajuća mobilna aplikacija sastoji se od klijentske strane bazirane na React Native Expo i serverskog dela sa Spring i REST servisima.

3.1. MagicDraw

MagicDraw je alat za modelovanje koji se prvenstveno koristi za dizajn i arhitekturu softvera. Podržava jezike za modelovanje kao što je UML (Unified Modeling Language). Široko se koristi u razvoju softvera za projektovanje arhitekture sistema, generisanje koda i upravljanje zahtevima. Stereotip u MagicDraw predstavlja mehanizam za proširenje standardnog UML metamodela. Primenom stereotipa, jezik modeliranja može se prilagoditi tako da bolje odgovara specifičnim potrebama projekta, pružajući više konteksta ili dodatne attribute elementima u dijagramu.

3.2. Apache Freemarker

Apache Freemarker je obrađivač šablona, odnosno Java biblioteka za generisanje tekstualnog sadržaja na osnovu šablona i promenljivih podataka. Šabloni su napisani u FreeMarker Template Language jeziku, koji je jednostavan i specijalizovan. Obično se za pripremu koristi programski jezik opšte namene, kao što je Java, za izdavanje upita u bazi podataka i obavljanje poslovne logike. Nakon toga, Freemarker prikazuje pripremljene podatke koristeći šablone.

3.3. REST

Representational State Transfer (REST) predstavlja stil softverske arhitekture koji definiše skup smernica za kreiranje pouzdanih stateless veb API-ja. Veb API koji se pridržava REST ograničenja opisuje se kao RESTful API. RESTful API se oslanja na HTTP protokol, pa se operacije nad resursima svode na HTTP metode kao što su GET, POST, PUT, DELETE i OPTIONS.

3.4. Spring

Spring je radni okvir napisan u Java programskom jeziku koji omogućava izgradnju REST arhitekture za Java aplikacije. Spring upravlja infrastrukturom tako da programer može da se fokusira na domenske probleme umesto na tehnologiju i realizaciju. U implementiranoj aplikaciji, paketi koji su primarno korišćeni sadržali su kontrolere, servise i repozitorijume.

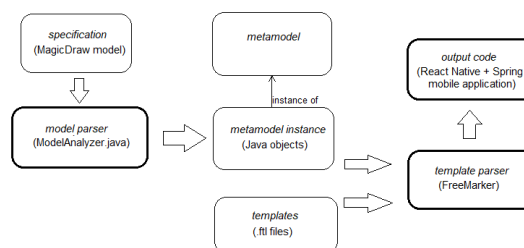
3.5. React Native Expo

React Native je radni okvir razvijen od strane Facebook-a koji omogućava programerima da kreiraju mobilne aplikacije koristeći JavaScript i React. Za razliku od tradicionalnih pristupa razvoju mobilnih aplikacija, koji zahtevaju odvojene baze koda za iOS i Android, React Native omogućava programerima da napišu jednu bazu koda koja radi na obe platforme. Expo je radni okvir i platforma izgrađena oko React Native koja dodatno pojednostavljuje proces razvoja pružanjem alata i usluga koje pomažu programerima da brzo izgrade, primene i iteriraju svoje mobilne aplikacije. Uključuje paket kao što

je Expo Go, mobilna aplikacija koja omogućava programerima da pregledaju svoje aplikacije na fizičkim uređajima bez potrebe za kompajliranjem koda.

4. ARHITEKTURA GENERATORA MOBILNIH APLIKACIJA

Odnos između MagicDraw alata i generatora koda u sistemu za generisanje mobilnih aplikacija može se videti na slici 1. Kao što je prikazano, izlaz iz MagicDraw alata je specifikirani model i koristi se kao ulaz za analizator, koji ga mapira na elemente generatora koji se podudaraju sa elementima meta-modela. Izlaz iz analizatora su Java objekti koji će se, zajedno sa šablonima, koristiti kao ulaz za Apache FreeMarker obrađivač šablona, iz kojeg se dobija izlazni kod mobilne aplikacije.

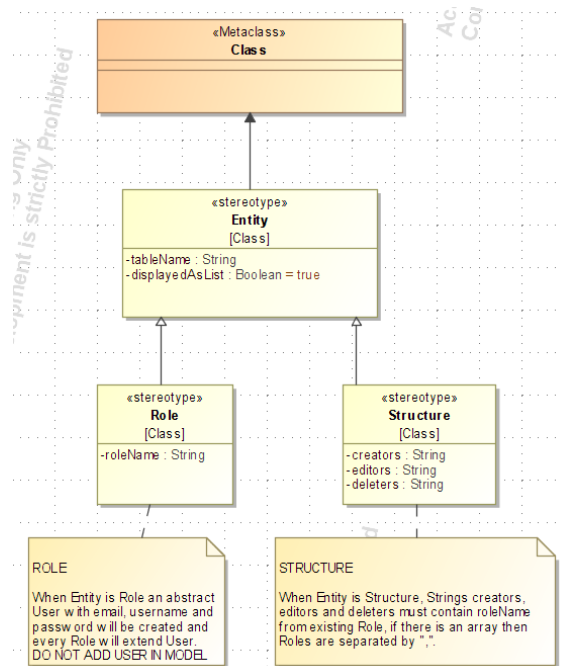


Slika 1 Arhitektura generatora mobilnih aplikacija

5. IMPLEMENTACIJA METAMODELA I MODELA

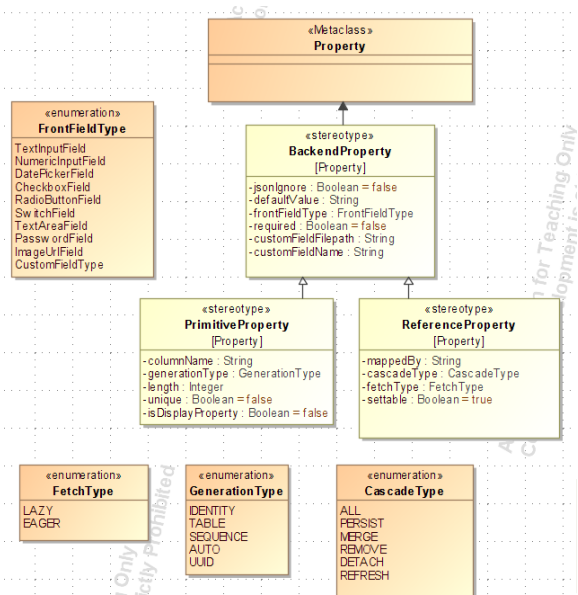
U kontekstu MDE, model je opis sistema ili dela sistema zapisan korišćenjem dobro definisanog jezika koji podržava automatsku interpretaciju od strane računara. Za kreiranje jezika primenjen je mehanizam koji koristi UML kao jezik opšte namene za prilagođavanje domenu primene. Proširenje UML-a može se sprovesti na više načina, a specijalizacija koja je primenjena u ovom radu je proširenje putem definisanja UML profila. Ova metoda podrazumeva definisanje stereotipa koji predstavljaju proširenje postojećih metaklasa, kao i definisanje označenih vrednosti, tagova koji predstavljaju meta-attribute stereotipa, i ograničenja koja mogu postavljati restrikcije na postojeća pravila jezika, ali ih ne mogu ukidati. Deo UML profila koji je korišćen za specifikaciju mobilne aplikacije može se videti na slici 2. U njemu su definisani stereotipi *Role* i *Structure*, koji nasleđuju *Entity* stereotip, koji predstavlja proširenje osnovne metaklase *Class*. Ako klasa predstavlja korisničku klasu, onda će se u modelu označiti sa *Role* stereotipom; u suprotnom, označiće se *Structure* stereotipom, u okviru kog se mogu definisati tipovi korisnika kojima je dozvoljeno kreiranje, uređivanje i brisanje te klase popunjavanjem tagova creators, editors i deleters. Ako ne postoje korisnici u modelu, te operacije su dostupne nad svim klasama koje su

prikazane, odnosno imaju *displayedAsList* tag označen sa true.



Slika 2 - UML profil za proširenje metaklase Class stereotipima Entity, Role i Structure

Na slici 3 je prikazan deo metamodela u kojem se proširuje metaklasa *Property* koja predstavlja apstrakciju promenljivih. Stereotipom *BackendProperty* se takođe definiše i tip polja koji će se koristiti u aplikaciji pri popunjavanju forme za kreiranje i uređivanje entiteta. Ovaj stereotip nasleđuje *PrimitiveProperty* stereotip, kojim se označavaju atributi koji su primitivnog tipa, i *ReferenceProperty*, kojim se označavaju atributi koji su tipa klase.

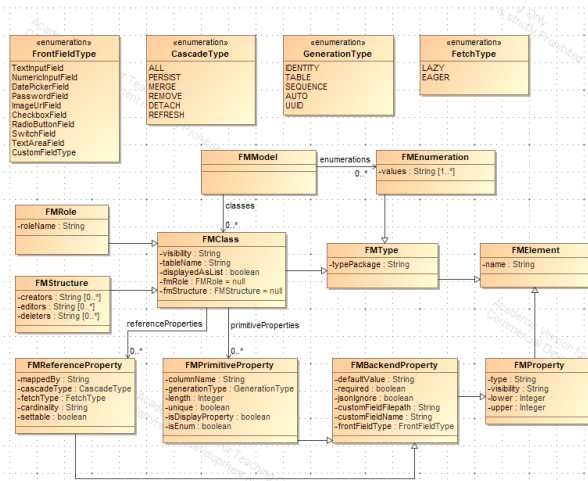


Slika 3 - UML profil za proširenje metaklase Property stereotipima BackendProperty, PrimitiveProperty i ReferenceProperty

6. IMPLEMENTACIJA GENERATORA KODA

Generator koda se najčešće implementira na bazi nekog obrađivača šablona. U ovom sistemu je razvijem

korišćenjem Apache FreeMarker-a pomoću kog su kreirani šabloni za: klase, kontrolere, servise, repozitorijume, enumeracije, kao i za fajlove na klijentskoj strani. Model sistema za generisanje koda se sastoji od 11 klasa i 4 enumeracije (Slika 4). Struktura klasa odgovara elementima kreiranog UML profila kako bi se podaci o elementima modela mogli mapirati na elemente generatora.

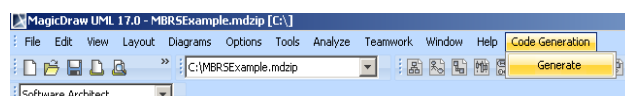


Slika 4 - Dijagram klasa generatora za prihvatanje podataka iz modela

Jedan od prvih koraka u radu generatora je analizirati dobijeni model aplikacije iz MagicDraw-a i mapirati sve podatke na strukturu podataka generatora optimizovanu za upotrebu od strane obrađivača šablona. Ova akcija se izvršava u klasi *ModelAnalyzer*. Analizator iterira kroz elemente paketa i izdvaja podatke koje skladišti u prethodno definisanim klasama i enumeracijama. Kada analizator završi svoj rad i svi podaci su namapirani na odgovarajuće strukture podataka, pokreće se generisanje koda. Ova funkcija je podeljena na više generatora koji su organizovani po vrsti fajla koji se generiše ili po podacima koji su potrebni u šablonima. Pri tome, svaki generator ima definisan naziv, lokaciju fajla u kojem se nalazi šablon za generisanje koda, naziv i tip fajla koji će biti izgenerisan, kao i lokaciju u aplikaciji gde će fajl biti sačuvan.

7. PRIKAZ RADA SISTEMA

Prvi korak u generisanju koda je kreiranje modela aplikacije u MagicDraw alatu i povezivanje sa postojećim meta-modelom kako bi se mogli koristiti stereotipi definisani unutar njega. Kada se to uradi, pokreće se Java projekat. Nakon toga automatski se pokreće MagicDraw alat u okviru kog će se pojaviti stavka u meniju sa nazivom *Code Generation* (Slika 5). Kada se klikne na *Generate* izgenerisaće se nov direktorijum u okviru Java projekta u okviru kog su dva podfoldera sa serverskom i klijentskom stranom mobilne aplikacije.



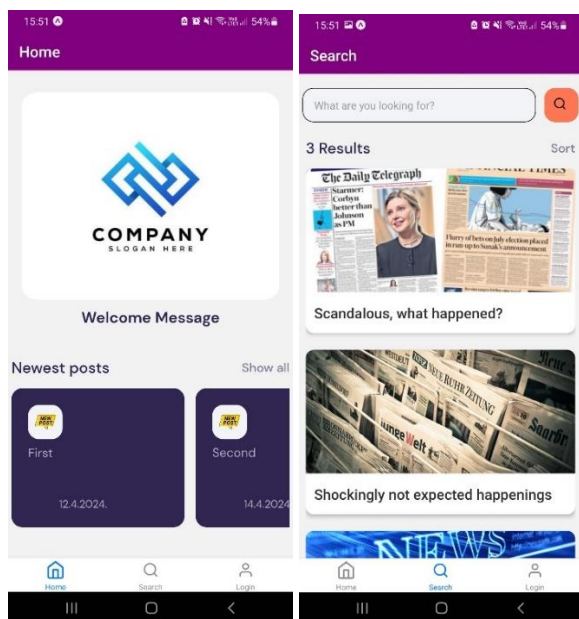
Slika 5 - MagicDraw meni sa stavkom Code Generation

Za pokretanje klijentske strane koristi se Expo Go mobilna aplikacija. Pošto je razvijena upotrebom React Native, omogućen je rad i na Android i na iOS operativnom

sistemu. Rad aplikacije će biti predstavljen korišćenjem aplikacije za blog objave. Sledeći tabovi su dostupni pri pokretanju mobilne aplikacije:

- *Home* tab (Slika 6(A)) – početni ekran koji se pojavljuje svim korisnicima, nezavisno od toga da li postoje korisničke uloge u modelu.
- *Login* tab – ekran za prijavu, postoji ukoliko su definisane korisničke uloge u modelu. Ispod prijave je moguće pristupiti stranici za registraciju klikom na Register.
- *Search* tab (Slika 6(B)) – ekran za pretragu nad listom, postoji nula ili više ukoliko se u modelu postavi meta-atribut *displayedAsList* na true nad klasom. U ovom primeru je samo nad *Post* klasom definisan *displayedAsList*. Klikom na jednu od kartica preusmerava se na nov prozor koji sadrži detalje o tom elementu.
- *Account* tab – nakon uspešne prijave *Login* tab se zamenjuje *Account* tabom. U okviru ovog prozora korisnik ima dugme da uredi svoje korisničke podatke, dugme za uređivanje podataka koje su jedinstveni po tipu korisnika, dugme za izmenu lozinke, dugme za brisanje naloga i dugme za logout.

Ukoliko postoje korisničke uloge u modelu, nakon logovanja će se u gornjem levom uglu ekrana pojaviti meni. U njemu se nalaze liste koje postoje u korisničkoj klasi. Takođe, tu su i liste objekata klase za koju je korisnička uloga definisana da može da kreira, uređuje ili briše, putem meta-atributa *creators*, *editors* i *deleters*. Ako ne postoje korisničke uloge u modelu, tada će *Search* tab sadržati dugmiće za brisanje objekta kartica i dugme za kreiranje novih objekata u donjem desnom ćošku ekrana. Kada se klikne na dugme, otvoriće se novi prozor sa formom koja sadrži sva polja definisana u modelu. Sva polja čiji meta-atribut *required* ima istinitu vrednost su označena zvezdicom u opisu, i ako se ne popune pre klika na *Save*, pojaviće se dijalog za upozorenje da nedostaje vrednost.



Slika 6 - (A) Home tab i (B) Search tab

8. ZAKLJUČAK

U ovom radu predstavljen je sistem za generisanje mobilnih aplikacija korišćenjem metoda inženjerstva softvera vođenog modelom. Glavna ideja inženjerstva vođenog modelom je da odvoji dizajn od tehničke implementacije, što programerima omogućava da se fokusiraju na koncepte iz domena problema, dok se automatski sistemi bave detaljima generisanja koda. Značaj ovog pristupa leži u tome što promovise ponovnu upotrebljivost, skalabilnost i održivost sistema, kao i brži razvoj. U radu su prikazani motivacija za razvoj sistema, korišćene tehnike i tehnologije. Prikazana je implementacija meta-modela u alatu MagicDraw, kao i njegov značaj pri kreiranju modela. Predstavljen je rad generatora koda napisanog u Javi, sa analizatorom za mapiranje elemenata definisanih u modelu na elemente generatora, dok je FreeMarker korišćen kao obrađivač šablona. Budući rad može uključivati proširenje meta-modela za organizaciju komponenti i stranica u mobilnoj aplikaciji, kao i unapređenje tehnologija korišćenih u razvoju.

9. LITERATURA

- [1] Apache Cordova, <https://cordova.apache.org/>, preuzeto septembra 2024.
- [2] Charland, Andre, and Brian Leroux. "Mobile application development: web vs. native." *Communications of the ACM* 54.5 (2011): 49-53.
- [3] OutSystems, <https://www.outsystems.com/>, preuzeto septembra 2024.
- [4] Gomes, Pedro M., and Miguel A. Brito. "Low-code development platforms: a descriptive study." *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2022.
- [5] Cervera, R. G., Esteban, D., & Garijo, F. J. (2010). "Model-driven development of mobile applications." *Journal of Systems and Software*, 83(10), 1715-1732.

Kratka biografija:



Olivera Mirilović rođena je u Novom Sadu 1999. godine. Diplomirala je na Fakultetu tehničkih nauka u Novom Sadu, na smeru Softversko inženjerstvo i informacione tehnologije 2022. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Softverskog inženjerstva o informacionih tehnologija odbranila je 2024.god.