

PRIMENA TEHNIKA MODELOM VOĐENOG RAZVOJA U RAZVOJU PLATFORMI ZA ONLAJN UČENJE

Ivana Spasojević¹, Miroslav Stefanović², Jelena Petrovački³, Dejana Gladić⁴, Andrea Gutai⁵

^{1,2,3,4,5}Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, Srbija

¹ivana.spasojevic@uns.ac.rs, ²mstef@uns.ac.rs, ³jelenapetrovacki@uns.ac.rs, ⁴dejanagladić@uns.ac.rs,
⁵gutaiaandrea@uns.ac.rs

Kratak sadržaj: Razvoj tehnologije i sve šire mogućnosti za njenu primenu dovode do potrebe za digitalizacijom u gotovo svim aspektima života. Ova potreba se javlja i u obrazovanju, a primarno se ogleda u težnji ka implementaciji i korišćenju platformi za onlajn učenje. U poslednjih nekoliko godina sve više univerziteta u upotrebu uvodi platforme za onlajn učenje, što rezultuje postojanjem velikog broja različitih platformi koje se svakodnevno koriste širom sveta. Većina platformi podržava osnovne nastavne procese kao što su postavljanje materijala, postavljanje obaveštenja studentima i organizovanje testova. Ipak, one su najčešće razvijene kao univerzalne platforme opšte namene i samim tim ne obuhvataju procese koji su karakteristični za domen nekog, konkretnog, obrazovnog područja. Problem koji se često javlja prilikom razvoja ovakvih sistema jeste otežana integracija različitih platformi, smanjena adaptivnost i uopšteno smanjena efikasnost rada. U ovom radu je dat prikaz upotrebe tehnika modelom vođenog razvoja u razvoju platformi za onlajn učenje. Upotreba ovih tehnika obezbeđuje prevazilaženje prethodno spomenutih problema, jer omogućava da se, zbog visokog nivoa apstrakcije, isti model nekog koncepta iz realnog sistema koristi u razvoju različitih platformi. Cilj upotrebe ovog pristupa se primarno ogleda u mogućnosti za značajno smanjenje kompleksnosti pri razvoju platformi uz istovremeno omogućavanje njihove višekratne upotrebljivosti.

Ključne reči: modelom vođen razvoj, MDD, platforme za onlajn učenje

USAGE OF MODEL-DRIVEN DEVELOPMENT IN DEVELOPMENT OF ONLINE LEARNING PLATFORMS

Abstract: The development of technology and the possibilities for its application lead to the need for digitalization in almost all aspects of life. This need also arises in education, and is primarily reflected in the aspiration to implement and use online learning platforms. In the last few years, more and more universities are introducing online learning platforms, which results in the existence of a large number of different platforms that are used every day around the world. Most platforms support basic teaching processes such as posting materials, posting notifications to students, and organizing tests. However, they are most often developed as general purpose platforms and thus do not include processes that are characteristic for the domain of a specific educational area. A problem that often occurs during the development of such systems is the difficult integration of different platforms, reduced adaptability and, generally, reduced work efficiency. This paper presents the use of Model-Driven Development techniques in the development of online learning platforms. The use of these techniques ensures overcoming the previously mentioned problems, since the high level of abstraction allows the same model to be used for modelling the same entity while developing different platforms. The goal of using this approach is primarily reflected in the ability to significantly reduce the complexity of platform development while enabling their reusability.

Key Words: Model-Driven Development, MDD, online learning platforms

1. UVOD

Potrebe za digitalizacijom gotovo svih aspekata života dovode do situacije da u današnje vreme postoji veliki broj tehnologija i platformi različitih namena koje su dostupne korisnicima. Isto tako, veoma često se javlja i potreba za unapređenjem postojećih platformi ili za razvojem potpuno novih. U situaciji kada se u okviru poslovnih sistema implementiraju kompleksni informacioni sistemi, njihova modifikacija ili unapređenje predstavlja veliki izazov. Prvi problem koji se javlja u spomenutoj situaciji jeste potreba za ulaganjem velike količine novca i vremena da bi se potrebna modifikacija realizovala. Takođe, kako bi se unapređenje sprovedo, neophodno je privremeno prekinuti većinu poslovnih procesa što automatski može uticati na poslovne rezultate.

U kontekstu obrazovanja, ovi problemi se javljaju prilikom razvoja platformi za onlajn učenje. Sve više univerziteta u upotrebu uvodi platforme za onlajn učenje, što rezultuje postojanjem velikog broja različitih platformi koje se svakodnevno koriste širom sveta. Neretko se visokoškolske ustanove suočavaju sa potrebom za unapređenjem platformi koje se trenutno koriste, njihovom integracijom sa nekim drugim platformama ili prelaskom sa jedne na drugu platformu [1]. Pored prethodno navedenih problema, integracija različitih platformi za onlajn učenje ili prelazak sa jedne na drugu platformu mogu dovesti i do trajnog gubitka podataka.

Kako bi se prevazišli spomenuti problemi sve češće se pristupa principima modelom vođenog razvoja softvera (engl. *Model-Driven Development* – MDD). Razvoj platformi za onlajn učenje upotrebom ovih principa se inicijalno svodi na definisanje modela kojima se reprezentuju osnovni entiteti i procesi iz realnog sistema, na najvišem nivou apstrakcije. Na taj način se obezbeđuje razdvajanje poslovne logike od detalja implementacije [2].

2. TEORIJSKE OSNOVE

Praćenje principa modelom vođenog razvoja se u današnje vreme smatra jednom od vodećih tehnika u razvoju softvera. Fokus je na razvoju modela domena kojim se opisuju osnovna ponašanja iz realnog sistema i sistemski podaci na različitim nivoima apstrakcije [1].

Principi modelom vođenog razvoja softvera se oslanjaju na predstavljanje domenskog znanja umesto znanja programiranja i algoritama. Modeli domena se vizuelno reprezentuju kombinacijom određenih grafičkih elemenata i veza između njih, čija notacija se definiše određenim jezikom za modelovanje.

2.1.1 Modeli

Modeli predstavljaju pojednostavljenje stvarnosti. Razvijaju se kako bi se obezbedilo lakše razumevanje realnog sistema, naročito ako se radi o kompleksnim sistemima. Prilikom modelovanja se koriste različiti pogledi na realan sistem, kako bi se on u potpunosti razumeo i kako se ne bi izostavili važni koncepti. Dešava se da različiti pogledi na sistem pružaju informacije o istom delu realnog sistema. U tom slučaju, dobijene informacije iz različitih pogleda moraju biti konzistentne. Osnovni princip koji se koristi prilikom razvoja modela je **apstrakcija**. Modeli se inicijalno razvijaju počevši od najvišeg nivoa apstrakcije, koji podrazumeva posmatranje samo osnovnih i najrelevantnijih informacija iz realnog sistema za problem koji se trenutno posmatra. Nivo apstrakcije se smanjuje uvođenjem detaljnijih informacija o modelu.

Jedna od ključnih prednosti primene modela i apstrakcije u razvoju informacionih sistema se ogleda u njihovoj nezavisnosti od platformi koje se koriste za implementaciju. Na ovaj način je omogućeno lako prevođenje modela, na primer, u različite programske jezike i samim tim su, u određenoj meri, rešeni problemi integracije različitih platformi ili unapređenja postojećih. Pored toga, velika prednost primene ovog pristupa u razvoju informacionih sistema se ogleda i u tome što nije potrebno poznavanje programskih jezika i koncepata programiranja kako bi se razumeli razvijeni modeli [3]. Umesto toga, modeli se često zapisuju putem nekog jezika za modelovanje.

2.1.2 Jezici namenski za domen

Jezici za modelovanje imaju notaciju čija primena omogućava verni prikaz koncepata iz realnog sistema. Ponekad se jezici za modelovanje razvijaju tako da u obzir uzimaju i koncepte koji se, specifično, javljaju samo u domenu primene nekog konkretnog sistema za koji se model kreira. Na primer, ukoliko se razvija informacioni sistem za visokoškolsku ustanovu koja kao obavezan deo nastavnih procesa na svim predmetima podrazumeva aktivnu saradnju studenata sa privredom, notacijom ovako razvijenih jezika će biti obuhvaćeni i procesi prikupljanja povratnih informacija o studentima iz privrede. Sa druge strane, razvoj modela upotrebom jezika opšte namene za visokoškolske ustanove čiji nastavni procesi se odvijaju na spomenut način ne bi pružio mogućnost modelovanja ovih, dodatnih, koncepata.

Jezici koji u obzir uzimaju koncepte karakteristične za domen nekog konkretnog domenskog problema nazivaju se jezici namenski za domen (engl. *Domain-Specific Modeling Language* – DSML). Osnovni cilj primene ovih jezika je povećanje nivoa apstrakcije u modelovanju na nivo koji je iznad nivoa opštih programskih jezika. Takođe, na ovaj način je omogućeno ekspertima u domenu realnog sistema za koje se razvija model da učestvuju u razvoju informacionog sistema [5].

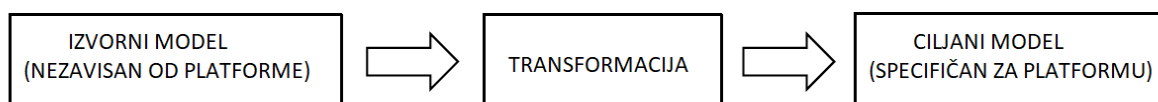
U visokoškolskim ustanovama, to su najčešće članovi nastavnog osoblja ili eksterni saradnici koji su usko specijalizovani za oblast koja se izučava u okviru visokoškolske ustanove. Oni su aktivno uključeni u svim fazama razvoja modela jer najbolje poznaju realan sistem, pa samim tim proaktivno pristupaju definisanju koncepata i njihovom povezivanju u logičke celine pri kreiranju modela koji će rešiti problem automatizacije u konkretnoj visokoškolskoj ustanovi. Pored eksperata u domenu, korisnici jezika namenskih za domen su i analitičari sistema i programeri koji razvijaju softver. Sve tri vrste korisnika analiziraju realan sistem iz različitog ugla, ali se svi oni vode istim zahtevima koje bi razvijeni model trebao da ispuni, a to su: jednostavnost, razumljivost i pogodnost za upotrebu.

Kako su poslovni sistemi izloženi stalnoj potrebi za prilagođavanjem promenama, veoma važno je obezbediti njihovu fleksibilnost. To se, takođe, može postići razvojem modela. Model sistema u svakom trenutku mora biti konzistentan tj. usklađen sa stvarnim stanjem u realnom sistemu. Iz tog razloga se često javlja potreba za uvođenjem izmena u inicijalni model. Ovaj proces se naziva *transformacija modela*.

2.1.3 Modelom vođen razvoj softvera i njegova primena u razvoju platformi za onlajn učenje

Modeli koji se inicijalno razvijaju se, kao što je već rečeno, razvijaju na najvišem nivou apstrakcije. Tako razvijeni modeli se nazivaju i *izvornim* modelima s obzirom na to da se radi o modelima koji su nezavisni od

platforme i programskih jezika u kojima se vrši implementacija. Zbog visokog nivoa apstrakcije ovi modeli se ne mogu koristiti u okviru razvojnih okruženja, već je potrebno izvršiti njihovu transformaciju u ciljane modele. Ovaj tok je prikazan na slici 1.



Slika 1. Proces transformacije izvornog modela u ciljani model [1]

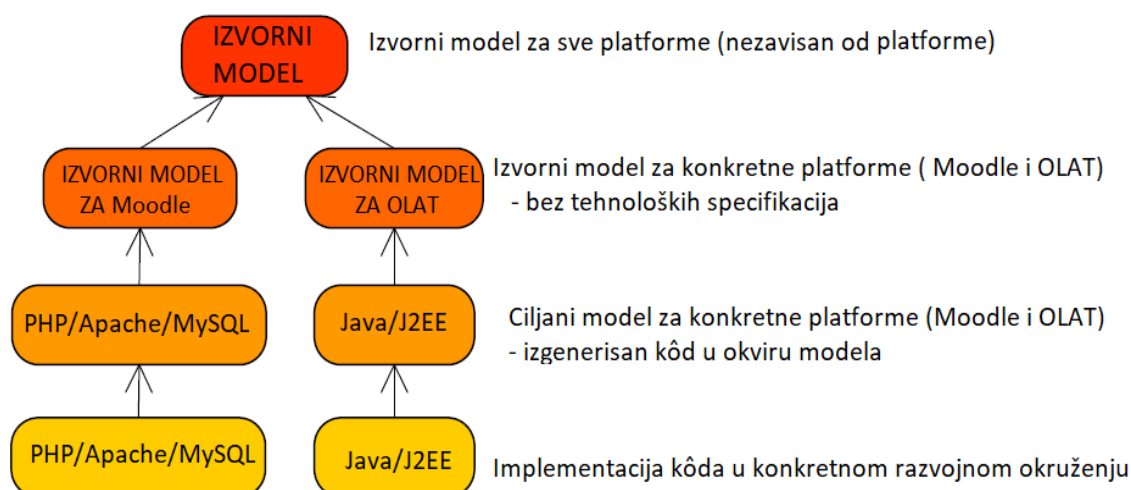
Ideja je da se obezbedi kreiranje izvornih modela primenom određenog jezika za modelovanje na osnovu kojih se, u procesu transformacije, može obezbediti automatsko generisanje implementacionog kôda u različitim programskim jezicima. Taj generisani kôd se zapravo smatra ciljanim modelom specifičnim za neku, konkretnu, platformu [7].

Proces transformacije se odvija automatski pomoću alata koji se naziva generator kôda. Ovaj alat prati određena pravila transformisanja, koja se primenjuju na izvorni model. Generator kôda najčešće vrši prevođenje tako što inicijalno generiše kôd za svaki koncept modela pojedinačno, dodeljujući mu pripadajuće osobine i veze. Nakon toga, generiše se kôd za svaki tip veze koji je korišćen u modelu. To znači da se prethodno izgenerisan kôd za pojedinačne koncepte na osnovu izgenerisanog kôda za veze integriše u jednu celinu [3]. Na primer, u izvornom modelu je definisano da u prvom koraku procesa izrade testa od strane studenta student Petar Petrović pokreće test Prvi test. Generator kôda prvo generiše kôd za entitete Petar Petrović i Prvi test, a nakon toga generiše kôd za vezu “pokreće”, čime je izvršena integracija resursa i procesa koji učestvuju u prvom koraku procesa izrade testa. Na ovaj način moguće je i da se, na primer, konkretnom studentu pošalje odgovarajuća poruka sa uputstvom o tome šta je i na koji način potrebno izvršiti. U kom programskom jeziku će biti generisan kôd zavisi od toga koja platforma za onlajn učenje se integriše na osnovu posmatranog modela.

Ukoliko se javi potreba za unapređenjem platforme razvijene nad izvornim modelom, ili njenom integracijom sa drugim platformama, dovoljno je izvršiti izmenu samog izvornog modela. Izmene izvornog modela se, ponovo, lako mogu implementirati praćenjem postupka sa slike 1. Isto tako, imajući u vidu

da se proces transformacije izvršava automatski, rizik od gubitka podataka u ovim procesima je minimalan [4].

Na slici 2 je prikazan primer upotrebe jednog izvornog modela platformi za onlajn učenje (razvijenog u nekom jeziku za modelovanje) za transformaciju u kôd napisan u različitim programskim jezicima.



Slika 2. Primer upotrebe jednog izvornog modela za implementaciju različitih platformi za onlajn učenje [6]

U konkretnom primeru su Bizonova i dr. primarno razvili opšti izvorni model, na najvišem nivou apstrakcije, za sve platforme za onlajn učenje [6]. Izvorni model se sastoji od koncepata kojima su predstavljeni elementi kurseva koji bi trebalo da se implementiraju na platformama. Nakon toga, izvršena je transformacija opšteg izvornog modela u izvorne modele za konkretne platforme (u ovom primeru Moodle i OLAT). Ova transformacija je izvršena proširivanjem inicijalnog modela kroz dodavanje koncepata specifičnih za domen konkretne platforme i obrazovnog područja za koji se model i razvija. Na taj način je primarni nivo apstrakcije modela smanjen. Sledeći korak jeste prevođenje izvornih modela za konkretne platforme u ciljani model, gde se na osnovu koncepata i veza izmodelovanih u prethodnom koraku generiše kôd u nekom konkretnom programskom jeziku. Poslednji korak u procesu prikazanom na slici 2 je implementacija izgenerisanog kôda u za to predviđenom razvojnom okruženju.

Neki od autora smatraju da je veoma važno prilikom razvoja modela platformi za onlajn učenje voditi računa o tome da modeli kurseva koji bi trebalo da postoje na platformama budu istovremeno orijentisani i na sadržaj i na aktivnosti. Modeli koji su orijentisani samo na sadržaj budućeg kursa, bez aktivnosti, obično ne obuhvataju pedagošku stranu onlajn učenja. Uključivanjem i aktivnosti u model se prevazilazi ovo ograničenje jer se tako može, na primer, definisati tok procesa učenja. Na taj način je nastavnom osoblju pružena mogućnost da prate aktivnosti i napredak studenata i da razviju strategije za poboljšanje uspešnosti studenata čiji intenzitet praćenja definisanog toka učenja nije zadovoljavajući [1].

3. ZAKLJUČAK

Kako bi se minimizirali troškovi kasnog otkrivanja problema i grešaka u procesu automatizacije poslovnih procesa veoma je značajno izvršiti detaljnu analizu resursa i procesa realnog sistema u okviru kog se vrši implementacija informacionog sistema. Jedan od najboljih načina za detaljnu specifikaciju koncepata realnog sistema jeste primena pristupa modelom vođenog razvoja softvera. Na ovaj način je, u kontekstu visokoškolskih ustanova, omogućeno detaljno modelovanje toka nastavnih procesa koje platforma za onlajn učenje treba da sadrži, kroz sekvencijalno navođenje aktivnosti i resursa koji se koriste u svakom od koraka. Vizuelno prikazivanje izmodelovanog toka se vrši primenom određenog namenskog jezika za domen, razvijenog za potrebe automatizacije procesa u nekoj konkretnoj visokoškolskoj ustanovi. Notacijom ovako razvijenih jezika su obuhvaćeni svi koncepti specifični za domen posmatranog realnog sistema čime se stvara mogućnost implementacije onlajn platforme za učenje koja obuhvata sve specifičnosti karakteristične za posmatrano obrazovno područje.

Kao što se može videti na primeru sa slike 2, ključna prednost razvoja platformi za onlajn učenje upotrebom tehnika modelom vođenog razvoja se ogleda u mogućnosti za prevođenje ideje o budućoj platformi u model date platforme bez razmišljanja o programskim jezicima u kojima će se implementacija izvršiti. Na taj način je omogućeno i uključivanje stručnjaka iz domena u proces razvoja modela, iako oni često nisu upoznati sa konceptima programiranja. Uzimajući u obzir da se modeli razvijaju nezavisno od platforme u okviru koje se vrši implementacija, rizik od gubitka podataka prilikom implementacije je sveden na minimum. Isto tako, važna prednost je i to što se unapređenja i modifikacije već implementiranih platformi za onlajn učenje vrše na nivou izvornog modela što znatno smanjuje mogućnost nastanka greške u ovom procesu, a istovremeno povećava njegovu efikasnost.

4. LITERATURA

- [1] Goran Savić, Milan Segedinac, *A model-driven approach to e-course management*, Australasian Journal of Educational Technology, 14-29, 2018.
- [2] Hong Wang, Dong Zhang, *MDA-based Development of E-Learning System*. 27th Annual International Computer Software and Applications Conference (COMPSAC'03), Dallas, Texas: IEEE, 684, 2003.
- [3] Juha-Pekka Tolvanen, Steven Kelly *Defining Domain-Specific Modeling Languages: Collected Experiences*, International Conference on Software Product Lines, Heidelberg, 198-209, 2005.
- [4] Marco Aurélio Graciotto Silva, Ellen Francine Barbosa, José Carlos Maldonado, *Model-driven development of learning objects*, 41st ASEE/IEEE Frontiers in Education Conference. Rapid City, SD: IEEE, 2011.
- [5] Xiaobo Li, Feng Yang, Yong-lin Lei, Wei-ping Wang, Yi-fan Zhu, *A model framework-based domain-specific composable modeling method for combat system effectiveness simulation*. Software and Systems Modeling (SoSyM), vol. 16, no. 4, 2017.
- [6] Zuzana Bizoňová, Daniel Ranc, Matilda Drozdova, *Model Driven E-learning Platform Integration*, 2nd PROLEARN Doctoral Consortium in Technology Enhanced Learning, Crete, Greece, 18. 09. 2008.
- [7] Zuzana Bizoňová, , Daniel Ranc, *Model Driven LMS Platform Integration*, The Third Advanced International Conference on Telecommunications (AICT'07), Morne, Mauritius, IEEE, 25, 2007.