Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/16TI01Vojinovic

NAMENSKI JEZIK I OKRUŽENJE ZA MODELOVANJE I SPECIFIKACIJU PROGRAMSKOG KODA ZA UPRAVLJANJE BESPILOTNIM LETELICAMA

A DOMAIN SPECIFIC LANGUAGE AND A FRAMEWORK FOR PROGRAMMING CODE MODELLING AND SPECIFICATION FOR A DRONE CONTROL

Veljko Vojinović, *Fakultet tehničkih nauka, Novi Sad*

Oblast –INFORMACIONI INŽENJERING

Kratak sadržaj – *U ovom radu predstavljen je način za modelovanje programiranja bespilotnih letelica. Osobe bez programerskog iskustva, uz pomoć tehnologija opisanih u radu imaće mogućnost potpuno samostalno da programiraju bespilotne letelice. U tu svrhu razvijen je namenski programski jezik za programiranje dronova „dronDsl“ i u njemu je definisana intuitivna sintaksa, pomoću koje krajnji korisnici mogu da programiraju dronove. Definisanje gramatike jezika, sintakse, editora programskog koda razvijano je u okruženju Epsilon i radnom okviru Xtext. Na osnovu definisane gramatike odnosno njenih pravila, korisnik izrađuje model koda za programiranje bespilotnih letelica u namenskom jeziku dronDsl, a zatim se taj model koda pomoću transformacija definisanih u ETL prevodi u izvršni kod Python radnog okvira PS Drone. Transformacije modela u model tzv. M2M transformacije, omogućavaju da osobe bez programerskog iskustva mogu da programiraju dronove isto kao i programeri. Zahvaljujući modelu koda pisanog intuitivnom lako razumljivom sintkasom i kasnije prevođenjem u zvanični radni okvir za programiranje dronova PS Drone postiže se da se dobija realan izvršni kod, spreman za programiranje dronova isto kao da je ispočetka pisan u radnom okviru PS Drone programskog jezika Python.*

Ključne reči: *DSL, metamodel, Ecore,bespilotne letelice, dron, Xtext, ETL.*

Abstract – *In this paper, a new method for programming drones is represented. Main idea is making possible for persons whoare not related to programming domain, to program drones individually, even without programming skills, thanks to technologies which will be described. To accomplish that goal a domain specific language was created dronDsl, and intuitive syntax was determined for persons without programming skills. Defining domain specific language, it's grammar, syntax, specific code editor was developed with Xtext and Epsilon framework. According to defined grammar rules, user creates model of programing drone in dronDsl, after model is created, it is transferred to executable Pythone code and framework PS Drone using ETL transformation technologies. Thanks to possibility of transferring model to model, non-coding personal can make a model of code in dsl and after, code will be transferred to official Python PS Drone code.*

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio doc. dr Milan Čeliković.

Keywords: *DSL, metamodel, Ecore, drone, DSL, Xtext, ETL, Python*

1. UVOD

Dronovi predstavljaju bespilotne letelice čija se upotreba prostire gotovo na sve aspekte života savremenog čoveka. Širok je dijapazon upotrebe dronova i njihovih mogućnosti, a neke od najkarakterističnijih primena jesu: vojno-izviđačka dejstva, zaštita životne sredine-monitoring i gašenje požara, prenos hrane i lekova na udaljene lokacije, geodetsko snimanje terena, infrastrukturni razvoj-snimanje nepristupačnih terena kao osnov za studije izvodljivosti i/ili sanacije uslova na terenu, turističko-rekreativne svrhe, zabavu-hobi i mnoge druge.Upotreba dronova je sveobuhvatna, te je zato i opravdana velika ekspanzija njihove proizvodnje u poslednjoj deceniji naše ere.
Mogućnosti i područja primene dronova, njihova realna potreba, nekad čak i životno važna i ekspanzija proizvodnje koja bi trebala biti izbalansirana sa brojem funkcionalnih-isprogramiranih dronova, iziskuju neophodnost postojanja softverskih rešenja koja to omogućavaju.U svetu se koriste različiti radni okviri (engl. *frameworks*) za programiranje dronova, međutim svaki od njih zahteva tehničku obučenost, odnosno takve okvire koriste programeri sa iskustvom. Budući da je upotreba dronova sve veća u svim granama industrije, a upotrebaljavaju se u velikom broju i u rekreativne svrhe, dronovi su sve više dostupni fizičkim licima, te je zbog toga potrebno da se programiranje dronova omogući i osobama bez programerskog znanja. U ovom radu dat je predlog rešenja za osobe bez programerskog iskustva koje žele da programiraju dronove. Savladavanje određenog programskog jezika opšte namene iziskuje vreme, trud i resurse. Budući da osobe bez programerskog iskustva dolaze iz sasvim različitih domena od programiranja i računarstva uopšte, često je ostvarenje tog cilja nemoguće u nekom kraćem vremenskom okviru. Zbog potreba posla, neprogramerskom osoblju bi bilo korisno da imaju jednostavan „alat“ koji im omogućava da prevedu kod koji intuitivno pišu bez programerskih veština u oficijalni kod koji se koristi u različitim programskim jezicima. U skladu sa time rad se bavi definisanjem namenskog jezika *dronDsl* za programiranje dronova.

2. POJAM BESPILOTNIH LETELICA

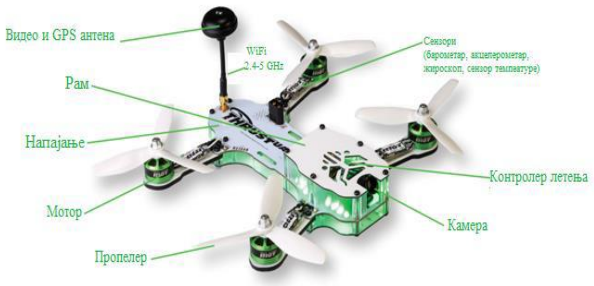
Bespilotne letelice su letelice koje su sposobne za kontinuirani let bez pilota (Bento M. F., 2008.). Bespilotne letelice su letelice ili avioni bez posade,

odnosno bez pilota, te mogu leteti samostalno sa unapred programiranim planom leta ili preko daljinskog upravljanja od strane pilota. U mogućnosti su da prenose različit teret i dolaze u različitim veličinama, od veličine ptice pa sve do veličine komercijalnih aviona. U poređenju s letelicama upravljanim od strane pilota, bespilotne letelice pružaju dve jako važne prednosti: ekonomične su i smanjuju rizik života pilota [1].

Komponente drona

Dron (Slika 1) je letelica s minimalno 3 rotirajuća tela (elise). Ova konfiguracija letelice naziva se Trikopter, i on je po načinu upravljanja drugačiji od ostalih multirotora kao što su kvadkopter (4 motora), hexkopter (6 motora), oktokopter (8 motora) itd. Jedino kod konstruktivne konfiguracije trikoptera jedan motor mora biti pomičan, kod svih ostalih konfiguracija motori su fiksni. Pojavom motora bez četkica (eng. *brushless*) i unapređenjem baterija, kao i minimizovanjem elektronskih komponenti došlo je do ekspanzije dron letelica na tržištu. Osnovni elementi drona su motori, propeleri, upravljač leta, baterija i konstrukcija koja sve to povezuje u celinu popularno nazvanu okvir (eng. *frame*).

Da bi dron mogao da bude upravljiv mora imati upravljač leta. To je elektronski uređaj koji u sebi ima ugrađen elektronski kompas, žiroskop, akcelerometar (merač ubrzanja) i često barometar. Svaki bolji upravljač ima i GPS (eng. *Global Positioning System*) modul za očitavanje položaja drona u prostoru. Obrađujući podatke od navedenih senzora upravljač šalje signale pojedinim motorima da smanje ili povećaju broj obrtaja, te na taj način održava stabilan let drona. Mnogi upravljači imaju mogućnost priključka eksternog-spoljnog senzora (npr. sonar), koji će dati podatak o udaljenosti od neke prepreke. Svi navedeni delovi drona smeštaju se u konstrukciju. Dronovi imaju mogućnost izvođenja programiranog leta. To znači da se putem programa unapred odrede tačke na koje treba dron da stigne sa definisanom visinom i brzinom između tačaka. Preciznost održavanja visine omogućava upravljač koji, kako je ranije navedeno, pored ostalih senzora, ima i barometar koji mu omogućava letenje na tačno zadatoj visini. Ovakav let se izvodi korišćenjem GPS podataka. Ukoliko na letelici imamo postavljenu kameru moguće je programirati let po tačno zadatoj ruti i na njoj definisati tačke na kojima će kamera snimiti fotografiju [4].



Slika 1. *Konstrukcija drona*

3. NAMENSKI JEZIK dronDsl

U prethodnim poglavljima smo videli da dronovi imaju značajnu ulogu u svakodnevnom životu savremenog čoveka. Naime, budući da upotreba dronova obuhvata mnoge sfere života i da umnogome olakšava posao u raznim granama privrede u okviru civilne upotrebe (u geodeziji, poljoprivredi,projektima infrastrukture, dostavi hrane i lekova, očuvanju životne sredine, spašavanju, gašenju požara), prepoznata je potreba za kreiranjem namenskog jezika za programiranje dronova *dronDsl*. Glavni zadatak kreiranja jezika jeste da osobe bez programerskog iskustva mogu uz upotrebu namenski kreirane sintakse uz par jednostavnih pravila pisanja koda, da isti taj kod mogu transformisati kasnije u izvršni kod popularnog radnog okvira pisanog u programskom jeziku Python pod nazivom PS-Drone koji se širom sveta koristi za programiranje bespilotnih letelica-dronova.

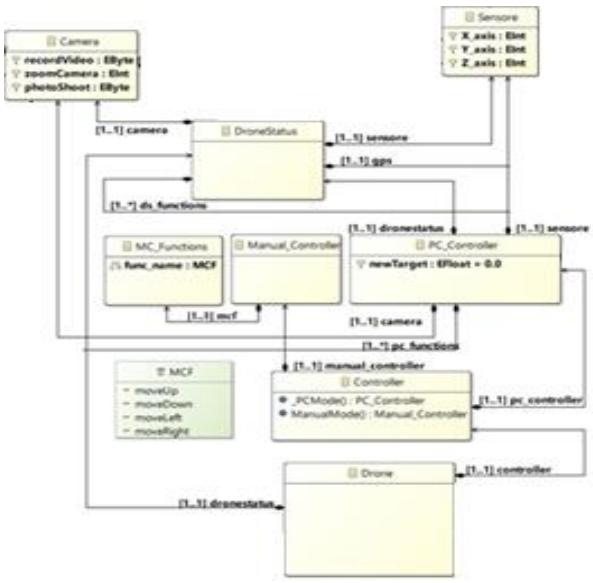
Na taj način primenom namenskog jezika za programiranje dronova postiže se da čak i osobe koje nisu u domenu programiranja mogu isprogramirati dronove shodno svojiim potrebama.
Osim toga, dodatan benefit se ogleda i u faktu da čak i programeri mogu iskoristiti ovaj način programiranja kako bi ubrzali proces programiranja dronova, budući da je sintaksa jednostavna, a osim toga programeri koji ne rade često sa Python programskim jezikom imaju pred sobom alat koji automatski za njih vrši transformaciju koda pisanog namenskim jezikom dronDsl u izvršni Python kod.

3.1. METODOLOGIJA IZRADE NAMENSKOG JEZIKA dronDsl

Polazna tačka pri izradi modela jeste modelovanje domenskog "objekta"-drona, u smislu modelovanja njegovih komponenti i njihovih funkcionalnosti. U tu svrhu napravljen je metamodel drona koji reprezentuje koncepte iz domena koji se modeluje. Nakon formiranja metamodela, treba doneti odluku o izboru tipa sintakse, koja može biti grafička ili tekstualna. Za programiranje dronova izabrana je tekstualna konkretna sintaksa, iz razloga što je krajnji cilj dobiti tekstualni izvršni *kod* koji se koristi za programiranje dronova u Python radnom okviru, kao i zbog drugih osobina koje tekstualna sintaksa poseduje.
Konkretna tekstualna sintaksa poseduje niz osobina koje se ogledaju u: mogućnosti bojenja ključnih reči definisanih gramatikom namenskog jezika, zatim u mogućnosti nadopunjavanja teksta (engl. *Auto content*), osim toga postoji veliki broj besplatno dostupnih alata za rad sa konkretnom sintaksom. Uzimajući u obzir navedene osobine tekstualnih sintaksi sa jedne strane I uzimajući u obzir činjenicu da se namenski jezik kreira za krajnje korisnike iz ne-programerskog domena,jasan je izbor tekstualne konkretne sintakse koja zbog svojih osobina značajno olakšava rad krajnjim korisnicima iz ne-programerske industrije.

3.2. APSTRAKTNA SINTAKSA

Metamodel reprezentuje apstraktnu sintkasu (Slika 2), na osnovu koje se mogu utvrditi njeni elementi – koncepti i odnosi među njima. Metamodel kao takav je dovoljan da se na osnovu njega izgeneriše automatski konkretna sintaksa,



Slika 2. *Metamodel namenskog jezika dronDsl*

međutim takav način kreiranja konkretne sintakse nije najpogodniji iz razloga što nije sasvim prilagođen domenu, te je potrebno nakon automatskog generisanja gramatike na osnovu metamodela izvršiti modifikaciju gramatike kako bi se prilagodila što više domenu. U tu svrhu izvršena je modifikacija početne gramatike tako da su definisana pravila pisanja koda u namenskom jeziku, takva da budu što jasnija krajnjem korisniku iz ne-programerskog domena. Potrebno je bilo kreirati što konciznije ključne reči, pravila sklapanja elemenata i dodatno metamodel je obogaćen OCL (engl. Object Constraint Language) ograničenjima koja su uvedena zbog kontrole sistema, npr. upotrebom OCL ograničenja sprečava se mogućnost unosa nedozvoljenih vrednosti, što dodatno povećava funkcionalost čitavog sistema odnosno namenskog pprogramskog jezika.

3.3. KONKRETNA SINTAKSA

Konkretna sintaksa namenskog jezika za programiranje dronova, dobijena je na osnovu kreiranog metamodela. Namenski jezik za programiranje dronova osmišljen je tako da sve funkcionalnosti drona budu ostvarene putem enumeracija, što znači da bi krajnji korisnik trebao da unese samo ime funkcije, npr. *loopTargetAndRecord* a kasnije se tako unet naziv funkcije transformiše u realnu funkciju Python programskog jezika i okvira PS Drone za programiranje dronova. *Xtext* poseduje namenski editor za kreirane jezike. Velika pogodnost takvih editora je mogućnost nadopunjavanja, tj. editor ima mogućnost prepoznavanja ključnih reči kao i ostalih elemenata konkretne sintakse, što praktično znači da je krajnjem korisniku dodatno olakšan rad. Pored jednostavne sintakse dodatno olakšanje upravo pruža osobina *Xtext* namenskog editora za prepoznavanje ključnih reči definisane gramatike, što je veoma bitno sa aspekta da je zamišljeno da će krajnji korisnici biti osobe bez programerskog iskustva. Za namenski jezik *dronDSL* početna verzija gramatike kreirana je na osnovu meta-modela upotrebom radnog okvira *Xtext*. Prilikom nerisanja gramatike, *Xtext* kreće od izabranog početnog koncepta i prati sve reference. Po jedno pravilo gramatike generiše se za svaki koncept na koji *Xtext* naiđe. Tako definisana sintaksa se treba preraditi kako bi odgovarala kreiranom metamodelu. Konkretna sintaksa namenskog jezika *dronDsl* koncipirana je tako da svaki koncept predstavlja zapravo jednu klasu sa atributima i funkcijama koje predstavljaju jedan konstruktivan element drona. Na primer koncept Camera ima funkcije phootoShoot, recordVideo i predstavlja konstruktivni deo drona s jedne, s druge strane to je i koncept metamodela , a i klasa koja se definiše kodom u namenskom jeziku. Sledeći opisani princip uspešno se mogu programirati dronovi namenskim jezikom *dronDsl*.

3.4. Primer modela pisanog namenskim jezikom dronDsl

U ovom poglavlju biće prikazan konkretan izgled jednog modela programiranja drona napisanog u namenskom programskom jeziku za programiranje dronova *dronDsl*. Dati primer predstavlja implementaciju namenskog jezika iz ugla krajnjeg korisnika. Na osnovu prethodno definisanog metamodela, pravila gramatike i sintakse korisnici koji nisu iz domena programiranja će moći na istovetan način kako je ovde prikazano da programiraju dronove. Dat je model koda napisanog u namenskom jeziku *dronDsl*. Drone, Camera, Gyroscope itd. predstavljaju koncepte metamodela i u okviru svakog koncepta postoje atributi i funkcionalnosti koje korisnik specificira kodom u zavisnosti od toga koje funkcionalnosti želi da omogući dronu. Na primer ako želi da promeni putanju kretanja drona zadaće novu vrednost koordinata newTarget u konceptu PC_Controller a ako želi da ga vrati na početnu poziciju poletanja pozvaće funkciju turnBackToStart.

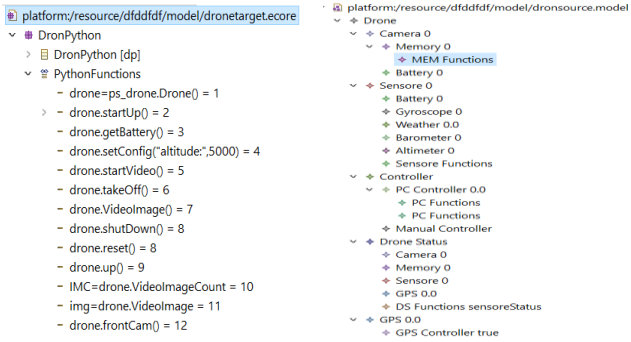
```
Drone {
 camera Camera
 { recordVideo 10 zoomCamera 25 photoShoot 25 }
memory Memory { memsize 100000
 MEM_Functions savePhoto saveVIdeo
checkFreeSpace memStatus } }
sensore Sensore { X_axis489222.57 Y_axis232475.78
battery Battery { level 50 batStatus }
 gyroscope Gyroscope { rotation-70 gyroStatus }
weather Weather { temp 22 time 12 windIntensity 15 }
PC_Controller {
        newTarget 1132457.00
        changeTarget
        loopTargetAndRecord
        secureHeight
        lostSignal
        turnBackToStart
```

4. M2M TRANSFORMACIJE KODA NAMENSKOG JEZIKA U PS.DRONE KOD

4.1. Izvorni i ciljni model

Izvorni model predstavlja dinamičku instancu prethodno kreiranog modela u namenskom jeziku dronDsl. Na slici 3 se može videti prikaz izvornog modela na kome se vide instance koju su nastale nakon korisničkog programiranja u namenskom jeziku i editoru.Ciljni model predstavlja Python PS-Drone model koda za programiranje dronova koji se koristi za komercijalno programiranje dronova. U ciljnom modelu se vide deklaracije i pozivanje funkcija koje se na istovetan način koriste u Pythonu. Cilj je da

izvorni model i njegovi elementi korespondiraju odgovarajućim funkcijama ciljnog modela, kako bismo „linkovali“ programski kod napisan namenskim jezikom *dronDsl* koji pišu osobe bez programerskog iskustva sa izvršnim Python kodom koji koriste programeri dronova.



Slika 3. *Izvorni i ciljni model M2M transformacije*

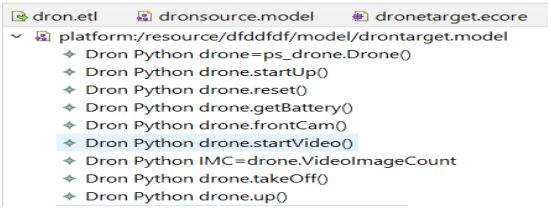
4.2.M2M transformacije

Glavni cilj projekta je transformisanje koda pisanog u namenskom programskom jeziku *dronDsl* sintaksom u PS-Drone radni okvir Python programskog jezika. Kako smo prethodno definisali izvorni i ciljni model transformacije, potrebno je upotrebom ETL-a i njegovih pravila transformacije izvršiti operaciju transformisanja M2M, odnosno prebacivanje iz modela u model, kako bismo od našeg modela pisanog u namenskom jeziku *dronDsl* dobili izvršni upotrebljiv kod za komercijalno programiranje dronova u Python PS Drone radnom okviru (Slika 4).

Nakon uspešno obavljenih transformacia dobijamo ciljni model izvršnog koda Python PS Drone okvira (Slika 5).

Slika 4. Transformacija izvronog u ciljni model

Slika 4. *Transformacija izvornog u ciljni model*



Slika 5. *Ciljni model- izvršni PS Drone Python kod*

5. ZAKLJUČAK

Namenski jezik *dronDsl*, omogućava samostalno programiranje bespilotnih letelica – dronova, krajnjim korisnicima koji nemaju konkretno programersko iskustvo. Jezik je napravljen tako da se vizuelno oslanja na metamodel drona, koji odgovara komponentalnoj strukturi drona , čime se povećava stepen intuitivnosti kod krajnjih korisnika, odnosno korisnik brže i lakše usvaja neophodno znanje za programiranje dronova u namenskom jeziku *dronDsl*. Fundamentalno jezik je osmišljen tako da metamodel predstavi konstruktivne elemente drona, koji se u okviru metamodela definišu kao koncepti i u sebi sadrže atribute i metode odnosno karakteristike i funkcionalnosti drona.

Definisana sintaksa jezika je intuitivna, sa bitnim aspektom a to su enumeracije funkcija. Namenski jezik za programiranje dronova dronDsl, pruža mogućnost navođenja imena konkretne funkcije bez definisanja, a kasnije se ta funkcija prevodi u oblik koji odgovara izvršnom modelu programiranja dronova sa potpunom definicijom. Na taj način korisnici indirektno programiraju dronove kao da sve vreme koriste Python i radni okvir PS Drone.

Dalji pravci razvoja namenskog jezika dronDsl mogu uključiti parcijalnu "automatizaciju" primenom veštačke inteligencije. Veštačka inteligencija bi umesto korisnika vršila pisanje koda u namenskom jeziku, na osnovu prepoznatih modela pisanja koda. Na taj način se dodatno olakšava programiranje osobama bez programerskog iskustva.

6. LITERATURA

[1] Fahlstrom P. G., Gleason T. J. (2012) *Introduction to UAV Systems.* Hoboken: Wiley

[2] Marcus Volter (2013) Designing, Implementing and UsingDomain-Specific Languages

[3] Žilić A. (2015), *Primjena bespilotnih letjelica u geodeziji na primjeru aerofotogrametrijskog sistema SenseFly eBee*, stručni rad. INZA d.o.o, Sarajevo.

[4] Žilić A. (2015), Primjena bespilotnih letjelica u geodeziji na primjeru aerofotogrametrijskog sistema SenseFly eBee, stručni rad. INZA d.o.o, Sarajevo.

[5] Felixge, (2017), node-ar-drone. Доступно на:https://github.com/felixge/node-ar-drone.

[6] Bryan V. (2014) Drone delivery: DHL 'parcelcopter' flies to German isle.Доступно на:http://www.reuters.com/article/us-deutsche-post-drones/drone-delivery-dhl-parcelcopter-flies-to-german-isle-idUSKCN0HJ1ED20140924

[7] Simmons D. (2016) Rwanda begins Zipline commercial drone deliveries Доступно на:http://www.bbc.com/news/technology-37646474

[8] https://edventures.com/blogs/stempower/drones-anatomy-101-getting-to-know-your-drone

Kratka biografija:

Veljko Vojinović rođen je u Požarevcu 1988. godine. Diplomirao je na Rudarsko-geološkom fakultetu 2012. godine, dok je master studije na istom fakultetu završio 2014. godine. Master studije, smer Informacioni inženjering upisuje na Fakultetu tehničkih nauka 2019. godine.