

НАМЕНСКИ ЈЕЗИК И ОКРУЖЕЊЕ ЗАСНОВАНО НА ИНЖЕЊЕРСТВУ ВОЂЕНОМ
МОДЕЛИМА ЗА ОПИСИВАЊЕ И ГЕНЕРИСАЊЕ БАСНИA DOMAIN-SPECIFIC LANGUAGE AND MODEL-DRIVEN ENVIRONMENT FOR
FABLES SPECIFICATION AND GENERATION

Сандра Бркић, Факултет техничких наука, Нови Сад

**Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И
ИНФОРМАТИКА**

Кратак садржај – У овом раду описан је графички наменски језик за моделовање басни за децу. Описане су технологије и окружење које је коришћено за имплементацију и дефинисање апстрактне синтаксе. са коришћеним концептима и значењима. Након тога дефинисана је конкретна синтакса графичког типа. Детаљно је описан коришћен алат и посупак креирања елемената и њихових алата. Приказан је и пример употребе свега дефинисаног у оквиру коришћених алата као и неопходна мапирања. Имплементирана су три генератора и начин њиховог функционисања, и то редом за сликовницу, текст и трећи који је комбинација претходна два. За реализацију рада коришћени су Eclipse Modelling Framework, Sirius и Xtend.

Кључне речи: басне, наменски језици, метамодел, развој софтвера базиран на моделу

Abstract – The paper describes a graphical domain-specific language for modeling fables for children. Used concepts and their meaning are presented, followed by concrete syntax and their graphic representations. The abstract syntax of this language, as well as detailed description of each of the concepts are described. The tools used and the process of creating elements and their tools are described in detail. A case study is presented. Three generators are implemented, for the picture book, the text, and the third one, which is a combination of the previous two. Technologies that are used are Eclipse Modelling Framework, Sirius and Xtend.

Keywords: fables, Domain-Specific Languages, metamodel, Model-Driven Software Development

1. УВОД

Басне су кратке приче за децу у којима су главни актери животиње. За њих су у главном везане људске особине као што је на пример лукавост, али и активности. Њихово понашање је типично људско.

Идеја је да прикажу карактерне особине, ситуације и односе међу људима, али тако да буду разумљиве деци у раном узрасту где је потребно да буду заинте-

НАПОМЕНА

Овај рад проистекао је из мастер рада чији ментор је био др Милан Челиковић, доцент.

ресовани необичностима, а животиње су увек интересантна тема.

Креатори оваквих прича често се сусрећу са проблемом креирања довољне количине различитих басни са сличним или разноликим улогама. Мотивација овог рада је да реши проблем смишљања што више оваквих прича за кратко време. Деца су константно заинтересована за још неку причу са њима драгим ликовима. Додатни проблем који се јавља је и културолошке природе, а тиче се тога да су неке животиње везане за одређену особине, док је на другом крају света то потпуно другачији случај. Ово доводи до креирања потпуно нове приче, а коришћењем овог наменског језика било би могуће само изменити тип животиње код једног лика.

Циљ рада је омогућење брже израде басни, а у случају недостатка маште или времена креатора, овај програмски језик намењен је ефикасном прављењу колекција басни са одређеним ликовима који би у њима учествовали. Додатно, овакав језик за циљ и могућност лаке модификације постојећих прича, као и креирање нових, а сличних постојећим. Он нуди брзо креирање, уз интуитивну употребу слика и предмета и уноса описа, како би текст касније био изгенерисан на основу њих.

Додатна предност настанка оваквог језика је и то, што би поред професионалних креатора, овако нешто могло да послужи и као додатна занимација деци одређеног узраста, која би била довољно велика да схвате коришћење доступних сличица.

Рад поред Увода и Закључка поседује још четири поглавља, од којих се у поглављу 2 говори о сличним решењима, поглављу три теоријским основама, док четврто представља коришћене технологије. У оквиру петог поглавља описан је процес имплементације наменског језика, а затим и самих генератора у поглављу шест.

**2. ПРЕГЛЕД ПОСТОЈЕЋЕГ СТАЊА У
ОБЛАСТИ**

Модел [1] представља издвајање битног од небитног, било да је у питању целокупни посматрани систем или само неки његов део. У току развоја софтвера они су се показали као јако корисни за пројектовање и даљу имплементацију. Модели се могу означити и као

поједностављење за разумевање неког проблема, или приказ потенцијалног решења.

Данас, најчешће употребљавани језик за моделовање је Unified Modeling Language (UML) [2], а могу се употребити на разне начине. Они могу послужити и као додатни начин комуникације између чланова тима, или било које заинтересоване стране, како би се на једноставнији начин представило функционисање система. За решавање проблема којим се бави овај рад до сад су сва постојећа решења заснована на AI(Artificial Intelligence) технологијама и не односе се на креирање комплетне приче већ на случајном избору имена и животиња за ликове или наслов, али не и целе приче.

3. ТЕОРИЈСКЕ ОСНОВЕ НАМЕНСКОГ ЈЕЗИКА И ТРАНСФОРМАЦИЈЕ МОДЕЛА

Језици наменски за домен (енгл. *Domain Specific Languages – DSL*) [1] су језици који дизајнирани тако да пруже решење за проблеме који припадају неком одређеном домену или области. Прилагођени су употреби у оквиру одређеног подручја, за разлику од општих програмских језика. Уколико су ови језици намењени за моделовање, онда се називају доменски специфични језици за моделовање.

Главне особине наменских програмских језика су:

- **Експресивност** — обликовани су тако да на најбољи могући начин искажу проблеме чијем решавању у одређеној области треба да допринесу.
- **Продуктивност** — употребом оваквих језика постиже се већа брзина у остваривању жељеног циља него употребом језика опште намене.
- **Поновна употребљивост** — имплементирано решење помоћу оваквих језика може се искористити поново.
- **Оптимизација** — пружају услуге оптимизације за специфичне задатке унутар домена ком су намењене.

Структура наменског језика прилагођена је домену за чију је употребу и намењена.

Основне компоненте наменског програмског језика су следеће:

1. **апстрактна синтакса** — служи да опише структуру језика и разумевање основних концепата који ће бити употребљавани. Треба да представи све неопходне компоненте у систему и проблему који треба језиком да се разреши, као и њихове могуће комбинације, везе и различите типове веза.
2. **конкретна синтакса** — представља репрезентацију језика моделовања. Ово је компонента која је намењена практичној употреби и која ће бити доступна крајњим корисницима. Постоје два типа ове синтаксе, а то су:
 - 1) **текстуална** — подаци се кодирају низом карактера, што је заступљено у програмским језицима и шире је распрострањено.

- 2) **графичка** — подаци се кодирају графичким елементима, њиховим распоредом, а могу се појавити и неки текстуални елементи.
3. **семантика** — ближе описује значење елемената који су дефинисани и начине на које их је могуће комбиновати и повезивати

Трансформација модела подразумева превођење из једног облика у други. Постоје два таква типа, а то су Model to Model(M2M) и Model to Text(M2T), који се користе зависно од тога шта је потребно урадити са моделима.

4. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА

Синтакса која је одабрана је графичког типа на основу метамодела, а трансформација се ради из модела у текст, односно M2T.

4.1. Eclipse Modelling Framework

Eclipse Modeling Framework (EMF) је главна технологија коју нуди Eclipse за моделовање, креирање метамодела. За то постоји посебни језик метамоделовања који се назива Ecoge. Такође у оквиру овог алата, постоји и могућност за креирање динамичке инстанце, као и за генерисање кода. Могуће је и изгенерисати неке друге алате који ће бити у даљој употреби за развој.

4.2. OCL

Object Constraint Language [4], или скраћено OCL је језик који служи за дефинисање додатних ограничења. Разлог за то је постојање ограничења која је потребно дефинисати у наменском језику, али је то због оскудности алата немогуће на нивоу метамодела, тако да је ово језик за дефинисање ограничења објеката у буквалном преводу.

Ова правила потребно је дефинисати на нивоу апстрактне синтаксе и у главном се примењују на класе, атрибуте, регеренце или односе између неких дефинисаних класа.

4.3. Sirius

Sirius [3] је пројекат Eclipse-а који нуди креирање едитора за графички језик коришћењем EMF – Eclipse технологије за моделовање. Коришћење овог пројекта првенствено је намењено за графички приказ и управљање моделом, а примену налази у графичкој репрезентацији објеката дефинисаних апстрактном синтаксом. Односно, служи за дефинисање конкретне графичке синтаксе неког програмског језика.

Алат обезбеђује спецификацију графичке синтаксе у облику неких графичких елемената. У оквиру Sirius-а није омогућено генерисање кода. Што се тиче репрезентација оне могу бити прилагођене потребама, различитим додавањем иконица или слика, а могу се користити и предефинисани облици који долазе са алатом. Оно што Sirius обезбеђује је могућност дефинисања едитора помоћу дијаграма, интегрисање апликација као и прилагођавање изгледа графичких елемената. За корисника нуди интуитивне едиторе и синхронизацију међу различитим едиторима.

4.4 Xtend

Xtend је програмски језик који је настао као алтернатива за Java програмере. Он је језик вишег нивоа и може да се извршава на Java виртуелној машини. Његова главна намена је да олакша развој Java апликација и да пружи лакшу и читљивију синтаксу од њене. Што се тиче извршавања, XTend код се преводи у чист Java код, даље се компајлира у java бајткод што на крају доводи до тога да је његова, и употреба Java програмског језика скоро иста. Битне карактеристике Xtend-а су краћа синтакса, функционални стил програмирања са проширеним могућностима и интероперабилност са Java-ом.

5. НАМЕНСКИ ЈЕЗИК

Наменски језик је језик који није језик широке употребе. Његова намена јесте да реши конкретан проблем и буде понуђена циљној групи корисника која се тим проблемом бави.

5.1. Апстрактна синтакса

Коришћена технологија за дефинисање апстрактне синтаксе наменског језика је EMF.

За сваки концепт биће дат опис, са називом, обележјима, семантиком као и разлозима за употребу. На слици 1 приказан је изглед дефинисаног метамодела. Коренски елемент назива се колекција басни (KolekcijaBasni), која представља сет басни које се заснивају на истој базичној причи, али могу имати исте учеснике.

Што се тиче коренског концепта за њега је битно дефинисати само назив, даље свака од басни која припада сету поседује свој наслов (Naslov), што је други битни концепт. У току сваког наслова постоји више сцена (Scena), које ће касније бити мапирани на пасусе у басни. Свака сцена, односно пасус, садржи ток елемената (Element) који могу бити радња (Radnja) или нека врста разговора (Dijalog). Додатно за концепт радње могуће је узети и концепте који се тичу времена (Vreme), као и места (Mesto) где се радња дешава.

5.2. OCL ограничења

У систему постоје ограничења која нису могла бити имплементирана на нивоу метамодела, него је то

морало да се уради на други начин. То је урађено помоћу OCL правила. Пошто се изабрани домен за наменски језик тиче басни, ту постоји велика слобода, тако да оваквих случајева није било много.

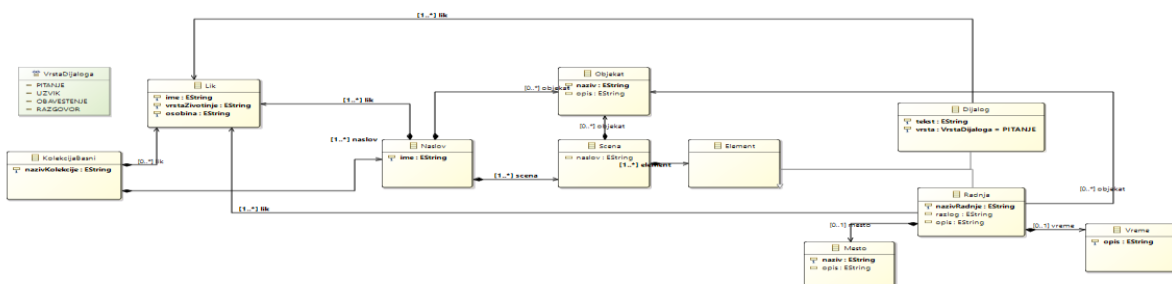
1. Оно што је путем OCL-а додатно дефинисано је да у оквиру колекције не могу постојати два Naslov-a, односно две басне с истим насловом.
2. Због случаја да сцене настају једна за другом, додато је правило и на самом концепту Scena, које се тиче провере јединствености њеног наслова.
3. Концепт Dijalog има посебно ограничење које се тиче броја учесника у разговору. Ограничење је такво да се он може водити између највише два учесника.

5.3. Конкретна синтакса

У оквиру Viewpoint-а предефинисано постоји један слој. Могуће је додати и више њих, али је за потребе овог језика био довољан један. На нивоу овог слоја могуће је дефинисати елементе као што су чворови и контејнери. Оно што је потребно урадити како би све функционисало како треба и да би се у пракси објекат појавио унутар свог контејнера је да се он дефинише угњешдено у контејнер. На овом нивоу потребно је и дефинисати све ивичне елементе који ће у графичкој синтакси бити репрезентовани као веза између нека два елемента.

Наредни корак био би дефинисање алата. За почетак је потребно додати секцију Section у оквиру овог Viewpoint-а и доделити јој неки назив, овде је то Alat. На овом нивоу потребно је за сваки претходно дефинисан елемент одабрати New Element Crea затим да ли је то чвор, контејнер или ивица.

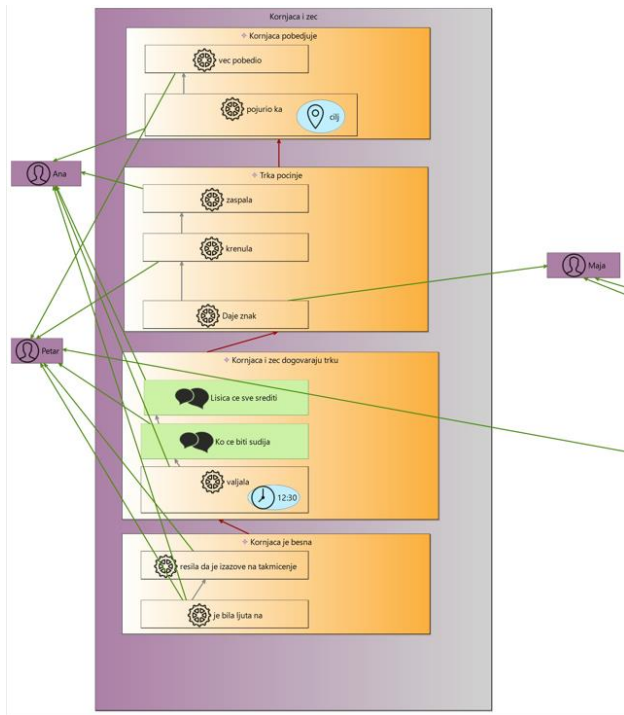
Иконице одабране за презентацију акција креирања нових инстанци горе побројаних концепата аналогне су онима којима се и репрезентују. С тим да су додате иконице и за контејнере за које у оквиру графичке представе није било потребе за приказивањем, као што су Naslov и Scena. На слици 2 приказан је пример написан дефинисаном конкретном синтаксом.



Слика 1 - Апстрактна синтакса

5.4. Пример употребе наменског језика

Пример са слике 2 приказује приче у оквиру које постоје ликови Маја, Ана и Петар. Представљена је једна приша из целокупне колекције која садржи четири сцене. У оквиру прве сцене налазе се две радње а за једну је дефинисано и место, друге три, док је трећа сцена испињена дијалозима и радњом з а коју је дефинисано време.



Слика 2 - Пример употребе

6. ОПИС ПРОЦЕСА ГЕНЕРИСАЊА ФАЈЛОВА

За генерисање фајлова који су у овом случају били Hypertext Markup Language (HTML) фајлови морало се извршити повезивање пројекта са Xtend форматом који је у могућности да препозна дефинисане концепте и приступи њиховим пољима. Покривено је генерисање сликовнице за млађе узрасте као и текстуалних прича за оне старије. Ово је извршено имплементацијом три генератора од којих је трећи комбинација претходна два.

Генератор функционише тако што пролази кроз дефинисане концепте и ради оно што се за њих специфицира. Потребно је конкретну синтаксу повезати са апстрактном да би генератор могао да их препознаје и генерише текст сходно типу на који наилази.

6.1. VasneGenerator имплементација

Прво се FOR петљом обезбедио пролазак кроз све наслове који у оквиру колекције постоје и за сваку је он исписан. За сваки наслов обезбеђен је пролаз кроз сцене које му припадају и такође исписан њихов назив. У овом случају није било потребе за приказом слике. Након тога било је потребно проћи кроз све елементе који се налазе у сцени, а који су могли бити радње и дијалози.

У овом случају било је потребно комплексније обрадити различите типове елемената и извршити потребне конверзије уколико су они припадали неком супертипу.

6.2. Print имплементација

Трећи генератор представља комбинацију претходна два. Његова сврха је да омогући раздвајање тако да свака прича има засебну нумерисану страницу у оквиру које ће бити приказан текст поред слике за коју је везан.

Измене које су биле потребне тичу се генерисања фолдера за колекцију и унутар њега засебних страница за приче. Што се тиче генерисања текста слика, коришћен је исти приступ као у претходна два генератора.

7. ЗАКЉУЧАК

Мотивација за израду оваквог језика јесте лакоћа креирања велике количине прича тако да имају исте ликове, како би се на основу тога створила колекција. Овим је деци могуће пружити већу количину смисленог садржаја на лак начин.

Овакав наменски језик могао би бити унапређен на много начина. У наставку су наведена три примера. Први од њих је могућност листања сликовнице где би било потребно креирати HTML странице са могућношћу листања и позиционирања на одговарајућу страницу. Додатно било би добро да буде омогућено генерисање pdf-а како би се садржај могао експортирати и у физички облик. На крају, као решење проблема који се јавио, а тиче се измена језичких конструкција ка падежима предлог је увезивање са неким од AI алата који би изгенерисани “неправилни” текст средило у одговарајућу форму.

8. ЛИТЕРАТУРА

- [1] Brambilla, M., Cabot, J., & Wimmer, M., “Model-driven software engineering in practice”, *Morgan Claypool Publishers*, 2012
- [2] <https://www.uml.org/> (приступљено у јулу 2024.)
- [3] Sirius - <https://www.eclipse.org/sirius/> (приступљено у јулу 2024.)
- [4] <https://www.omg.org/spec/OCL/2.4/About-OCL> (приступљено у јулу 2024.)

Кратка биографија:



Сандра Бркић рођена је 9. маја 2000. године у Београду. Завршила је природно-математички смер „Шабачке гимназије“. Школске 2019/2020 уписује Факултет техничких наука у Новом Саду, на студијском програму Рачунарство и аутоматика. Основне студије завршава 2023 године, када уписује мастер програм Примењене рачунарске науке – Инжењеринг информacionих система.