

JEZIK ZA SPECIFIKACIJU KVIZOVA I UPITNIKA**A SPECIFICATION LANGUAGE FOR QUIZZES AND SURVEYS**Nevena Alavanja, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – Uz oslonac na meta-jezik *textX*, dizajniran je i implementiran jezik specifičan za domen specifikacije kvizova i upitnika. Implementirane su i frontend i backend aplikacije za učitavanje i ispunjavanje kvizova i upitnika opisanih ovim jezikom. Rad pruža pregled teorijskih osnova i tekućeg stanja u oblasti, kao i detaljan opis rešenja i studije slučaja.

Ključne reči: jezici specifični za domen, *textX*, parser, kviz, upitnik

Abstract – Relying on the meta-language *textX*, a domain-specific language for specifying quizzes and surveys has been designed and implemented. Both frontend and backend applications have been implemented for loading and completing quizzes and surveys described in this language. This paper provides an overview of the theoretical foundations and current state in this field, as well as a detailed description of implemented solution and case studies.

Keywords: domain-specific languages, *textX*, parser, quiz, survey

1. UVOD

Svakodnevno, kvizovi, testovi, ankete, upitnici i slične forme procena su u upotrebi u najrazličitijim domenima i u razne svrhe. Oni su alat za prikupljanje vrednih informacija, procenjivanje znanja, mišljenja, stavova, donošenje informisanih odluka, utvrđivanje potreba, prilagođavanje proizvoda i usluga potrebama ciljne grupe ili prosto, ponekad, i vid zabave. Kvizovi i testovi imaju najznačajniju ulogu u domenu obrazovanja, dok su upitnici/ankete neizostavan deo svake oblasti koja u nekom trenutku razvije potrebu za prikupljanjem podataka i povratnih informacija od strane korisnika, zaposlenih, stanovnika, ili bilo koje druge ciljne grupe. Kako su njihovi domeni upotrebe najraznovrsniji (edukacija, politika, sociologija, marketing itd.), tako su i načini na koji se ove procene sprovode različiti od slučaja do slučaja i zavise kako od mogućnosti i preferencije, tako i od same ciljne grupe. Uzimajući u obzir razvoj tehnologije i sve veću primenu interneta u razne svrhe, može se pretpostaviti da se i sa sprovođenja kvizova i upitnika na papiru, uživo – „na terenu”, preko telefona i sličnih načina, prešlo na sprovođenje u najvećoj meri putem raznih internet platformi.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Igor Dejanović, red. prof.

Shodno tome, ovaj rad ima za cilj istraživanje postojećih softverskih rešenja za što lakše kreiranje ovakvih procena, kao i implementaciju jednog takvog.

Kao deo rada, razvijen je **JSD** (jezik specifičan za domen) čija je osnovna karakteristika to što se prilagođava domenskim stručnjacima, i što njegova sintaksa teži ka tome da bude razumljiva i osobama koje nisu iz sveta programiranja. Osim jezika specifičnog za domen (opisanog pomoću meta-jezika **textX**), rešenje čine i parser jezika (takođe dobijen pomoću *textX*), server koji transformiše i dostavlja generisan model opisan ovim jezikom, i *frontend* aplikacija koja služi kao korisnički interfejs za učitavanje i učestvovanje u kvizu ili anketi.

2. KLASIFIKACIJA PROCENA

Kako će u nastavku biti pomenute razne vrste procena (engl. *assessment*), bitno je na samom početku razumeti njihovu svrhu, formu, osnovne karakteristike i razlike i relacije između njih.

• **Kviz** – najkraća, najčešće korišćena i najmanje formalna forma evaluacije od moguće tri - kviz, test i ispit. Ovaj termin je najčešće korišćen za neformalne i kraće testove, ili npr. nenajavljene testove znanja (engl. *pop quiz*). Ponekad, on može da se koristi za procene zabavnog tipa, bez ozbiljnijih posledica (pab kviz, onlajn kviz, TV kviz). Pojedini kvizovi nemaju bodovanje, već mogu služiti samo kao procena znanja i materijala kako bi se identifikovale oblasti koje je potrebno doučiti. Pitanja u kvizu mogu biti raznovrsna kad je reč o njihovom tipu: popunjavanja praznina, pitanja sa jednim ili više tačnih izbora kao odgovor, *open-ended* pitanja, *true/false* pitanja itd.

• **Anketa** (engl. *survey*) – najčešće korišćena forma i tehnika za prikupljanje podataka o određenoj temi. Ankete su neophodne za dobijanje ključnih podataka za razvoj efikasnih strategija i zadovoljstvo zaposlenih, klijenata ili kupaca. Mogu se sprovesti na više načina – na papiru, onlajn, preko telefonskog poziva ili intervjuom uživo. Da bi anketa bila uspešna, bitno je postavljati jasna, koncizna, nepristrasna i objektivna pitanja. Onlajn anketa je u obliku forme koju korisnik popunjava, a ista se može distribuirati preko društvenih mreža, email-a, QR kodova ili URL-a [1]. Ovom vrstom ankete se štedi na vremenu i olakšava kasniji pregled i obrada podataka. Osim toga, dodatne prednosti su: tačnost, lakoća učešća za korisnike, dodatan branding koji kompanija postiže korišćenjem logoa i ostalih obeležja, postojanje već gotovih šablona (engl. *templates*) koji se mogu iskoristiti itd.

• **Bodovana anketa** (engl. *scored survey*) – anketa u kojoj svaki odgovor nosi određeni, fiksni broj bodova. U ovom tipu procene, kao ni kod obične ankete, ne postoje tačni i

pogrešni odgovori, već se na osnovu sakupljenih bodova formira određeni rezultat ankete za svakog učesnika [1]. Sabiranjem bodova korisnikovih odgovora dobija se konačan broj bodova, na osnovu koga se prikazuje rezultat u formi poruke. Obzirom na strukturu bodovane ankete, ona se može koristiti i u svrhu kviza, testa ili procene znanja, ali se najčešće praktikuje kao sredstvo za kreiranje testova ličnosti i sličnih procena zabavnog karaktera.

- **Kratka anketa** – (engl. *poll*) – anketa sa tačno jednim pitanjem, koja za cilj ima brzo prikupljanje povratne informacije (engl. *feedback*), preferencija, mišljenja itd.

3. POSTOJEĆA REŠENJA

SLang – tekstualan jezik specifičan za domen za kreiranje upitnika/anketa (engl. *Survey Questionnaires*). SLang ne razmatra prezentaciju upitnika i korisnički interfejs (engl. *User Interface* – UI), kao ni opšte aspekte koji se odnose na sisteme za prikupljanje podataka, kao što su bezbednost i analiza podataka [2]. Dizajn ovog jezika uključuje dve vrste transformacija modela u tekst (engl. *models to text* - M2T): transformaciju upitnika u SQL šemu i transformaciju u softver za prikupljanje meta-podataka koristeći JSON notaciju.

LiaScript - jezik specifičan za domen za kreiranje interaktivnih onlajn kurseva. Dizajniran je da bude kompatibilan sa *Common-Markdown*-om, ali takođe uvodi nekoliko jezičkih ekscentričnosti za rad sa kvizovima, anketama, *text2speech*, animacijama, ASCII-umetnošću, integraciju sa JavaScript-om itd. Ovaj jezik ima sopstveni makro-sistem koji pojednostavljuje dosadne i ponavljajuće zadatke sa uključenim sopstvenim *just-in-time* kompajlerom koji radi u pretraživaču i zbog toga ne zahteva dodatne alate [3]. LiaScript je interpreter koji preuzima i renderuje originalni *Markdown* dokument direktno unutar pretraživača.

Questionnaire DSL je jezik specifičan za domen (opisan pomoću *textX*-a) za opisivanje upitnika koristeći jednostavan interpreter terminala. Ovaj paket obezbeđuje interpreter upitnika koji prima model kreiran parsiranjem upitnika (opisanog ovim jezikom) i opciono kreira rečnik podrazumevanih odgovora identifikovanim dodeljenim ID-em ili rednim brojem pitanja.

TypeForm – alat za kreiranje *people-friendly* formi i anketa, koristeći neki od mnogih šablona (engl. *template*) u ponudi. Neki od šablona su šabloni za formu, kviz, anketu, kratku anketu, registraciju, izveštaj itd. Šabloni su takođe usmereni ka raznim domenima i predefinisanim temama i shodno tome dizajnirani (slike, raspored, boje, fontovi itd.) Nakon izbora odgovarajućeg šablona, ovaj alat omogućava izmenu i prilagođavanje istog izmenom teksta pitanja, tipa pitanja (*multiple choice*, *contact info*, *address*, *short text*, *long text*, *ranking*, *yes/no*, *opinion scale*, *rating*, *date* i mnogi drugi), slika, boja, logike itd. Nakon kreiranja forme, kviza, ili druge procene, ista se može sačuvati a zatim podeliti preko URL-a ili integrisati sa nekom od mogućih opcija.

TestPortal predstavlja drugo rešenje, veoma slično prethodnom alatu, osim što je fokusiran isključivo na procene koje proizvode određeni rezultat, bez podrške za upitnike i ankete.

ProProfs alat nudi najviše opcija, kao i najveći nivo prilagođavanja i podešavanja opcija: broj mogućih pokušaja, dozvola zabrana čuvanja i nastavljanja već započetog, vremensko ograničavanje, mogućnost premotavanja videa, automatsko pokretanje videa, redosled pitanja, omogućavanje preskakanja videa itd.

Zajednička karakteristika svih pomenutih onlajn alata jeste to što je za iole veću kontrolu kao i pregled rezultata i statistike neophodno pretplatiti se na *enterprise*, poslovni ili premijum plan.

Osim toga, kreiranjem procena putem nekog od pomenutih alata, dobiće se samo gotov proizvod, tj. kreiran kviz ili upitnik dostupan samo putem linka, bez slobode i mogućnosti da se potencijalno iskoristi ili integriše u neko softversko rešenje kao deo veće aplikacije.

Takođe, nema mogućnosti obrađivanja podataka tj. pristiglih odgovora u nekoj internoj bazi podataka, već su oni samo vidljivi putem ovih alata koji se u ovom konkretnom slučaju plaćaju.

4. JEZICI SPECIFIČNI ZA DOMEN

Fowler jezike specifične za domen definiše kao programske jezike ograničene ekspresivnosti fokusirane na određenu oblast. On naglašava da ekspresivnost zapravo proizilazi iz fokusiranja na uzak domen [4].

Takođe su opisani i kao programski jezici ili izvršive specifikacije jezika koje pružaju, kroz prikladne notacije i apstrakcije, široku ekspresivnost fokusiranu, a često i ograničenu, na specifičan domen problema [3].

JSD se nazivaju još i mali jezici (engl. *little languages*), a uspešnim JSD se smatra onaj koji je fokusiran na uzak, dobro definisan domen i pokriva ga na odgovarajući način.

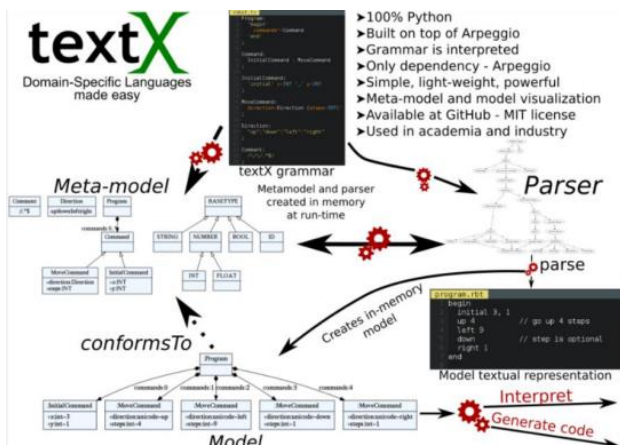
Neki od poznatih jezika specifičnih za domen jesu SQL, build jezici - Ant/Maven/Gradle, poslovni procesi - BPMN, DOT, PlantUML, Sed, HTML, CSS, XML, UML itd. [4].

5. TextX

TextX je meta-jezik (jezik za definisanje jezika) i alat za kreiranje jezika specifičnih za domen u programskom jeziku **Python**. Izgrađen je na osnovu **Arpeggio PEG** parsera, i koristeći njega, kreator jezika izbegava ručno konvertovanje stabla parsiranja u apstraktnu reprezentaciju [5]. Suštinski, doprinosi lakoj i jednostavnoj izgradnji potpuno novog tekstualnog jezika ili izgradnji podrške za već postojeći jezik.

Iz samo opisa gramatike, *textX* će automatski kreirati meta-model (u formi Python klase) kao i Arpeggio parser za opisani jezik. Parser će parsirati izraze kreiranog jezika i automatski izgraditi graf Python objekata (model) u odnosu na dati meta-model.

Dobijeni model može se dalje interpretirati, može se generisati kôd ili vršiti razne analize i transformacije (slika 1).



Slika 1. Arhitektura textX-a [6]

6. OPIS REŠENJA

Rešenje tj. projekat, sastoji se iz sledećih komponenti:

- **Jezik** i njegova **gramatika** opisana meta-jezikom textX
- **Parser** – izgrađen od strane textX-a
- **Server** - Uvicorn veb server implementiran za Python
- **Frontend** aplikacija – ReactJS aplikacija za prikaz korisničkog interfejsa kreiranih kvizova i upitnika

6.1. Gramatika

Gramatika jezika osmišljena je uzevši u obzir to da će ovaj jezik, kao i svaki JSD, potencijalno koristiti i osobe koje nisu iz programerske struke i nemaju tehničko predznanje već samo domensko. Shodno tome, težilo se ka lako razumljivom i čitljivom jeziku, bez gomilanja nepotrebnih simbola kao što su oble, vitičaste ili uglaste zagrade, dvotačke i sl. Može se reći da je kreirani JSD najbliži programskom jeziku Python po svojoj sintaksi, sa kojim deli i činjenicu da se kôd za veću čitljivost može pisati nazubljeno, ali nije neophodno. U nastavku ovog poglavlja, biće pojašnjena neka od pravila gramatike, opisana već pomenutim meta-jezikom textX.

Procena (engl. *Assessment*) je prvo i glavno pravilo gramatike ovog jezika, i ono obuhvata sve ostale elemente neophodne za kreiranje bilo kojeg tipa procene: naslov (engl. *title*), opis (engl. *description*), opcija za postavljanje pitanja za prikupljanje ličnih podataka (engl. *ask_for_personal_info*) i detalje procene (engl. *assessment_details*).

Detalji procene definisani su pravilom *AssessmentType* koje je definisano iskazom izbora, te označava da polje *assessment_details* može biti definisano nekim od sledećih pravila: *Quiz*, *Survey*, *ScoredSurvey*, *Poll*.

Kviz (engl. *Quiz*) pravilo čine: tip procene (engl. *type*), lista pitanja (engl. *questions*), kraj procene (engl. *end*), podrazumevani poeni pitanja (engl. *default_points_per_question*), podrazumevani negativni poeni pitanja (engl. *default_negative_points_per_question*) i kriterijumi polaganja (engl. *pass_criteria*).

Kviz pitanje (pravilo *QuizQuestion*) definiše se navođenjem karaktera "Q" i dvotačke, nakon čega, u proizvoljnom redosledu, slede: tekst pitanja, detalji pitanja (polje *question_details* definisano pravilom *QuizQuestionType*) i poeni pitanja (polje *question_points* definisano pravilom *QuestionPoints*).

Pitanja kviza i anketa mogu biti opisana nekim od sledećih pravila: *OpenEndedQuestion*, *SingleChoiceQuestion*, *MultipleChoiceQuestion*, *TrueFalseQuestion*, *NumberQuestion*, *OpinionScaleQuestion* i *RatingQuestion*.

Pravilo *Answer* opisuje odgovor na pitanje bilo kog tipa. Ono se opisuje na sledeći način: crtica (-), uglaste zagrade koje mogu biti prazne, sadržati informacije o tome da li je odgovor tačan (navodi se karakter "x") ili sadržati informacije o broju poena koje nosi odgovor (navodi se kao decimalan ili ceo broj). Nakon zagrada, sledi string koji opisuje tekst pitanja.

6.2. Parser

Kako bi se dobili svi relevantni podaci o jeziku, sve klase jezika i parser koji će biti u stanju da čita programe na datom jeziku, kreira se **meta-model**. Meta-model kreiraće se u ovom rešenju pomoću Python funkcije *meta-model_from_file*. Osim ovoga, registrovani su **procesori** koji vrše određene validacije i modifikacije nad prosleđenim objektima (npr. da li su poeni pitanja ili odgovora navedeni u slučaju da su obavezni, nisu navedeni ukoliko nisu dozvoljeni, koliko je tačnih odgovora definisano, da li je bodovanje kviza ispravno itd.).

6.3. Server

Backend deo ovog rešenja čine veb server i jedan API koji služi za dostavljanje modela zahtevanog kviza ili ankete od strane *frontend*-a. Podaci se pre slanja na klijentsku aplikaciju oblikuju pomoću DTO klasa.

Za kreiranje API-ja korišćeno je radno okruženje (engl. *framework*) FastAPI. REST API koji se koristi za dobavljanje resursa sa backend na frontend ima url "assessment", a metoda zahteva je GET. Nakon prijema zahteva, koji u *query* parametrima sadrži ime ili putanju fajla opisane procene, u ovoj funkciji se generiše model, transformiše u odgovarajući oblik pomenutih DTO klasa, a zatim enkoduje u JSON format, i kao takav response šalje. Veb server koji je korišćen u ovom rešenju je Uvicorn. Takođe, omogućeno je pokretanje programa preko komandne linije (cmd), sa opcijama "--run_server" i "--assessment_file" koje mogu kreirati meta-model (i njegovu vizuelizaciju) i model, bez pokretanja servera.

6.4. Frontend

Za potrebe korisničkog interfejsa, razvijena je React aplikacija koja predstavlja alat preko kojeg se može izabrati određena procena, učitati a zatim odgovoriti na sva pitanja i videti rezultat/broj bodova ukoliko postoji. Postoje tri glavne stranice: stranica za izbor fajla procene, stranica sa podacima i pitanjima kviza ili upitnika i stranica sa rezultatima.

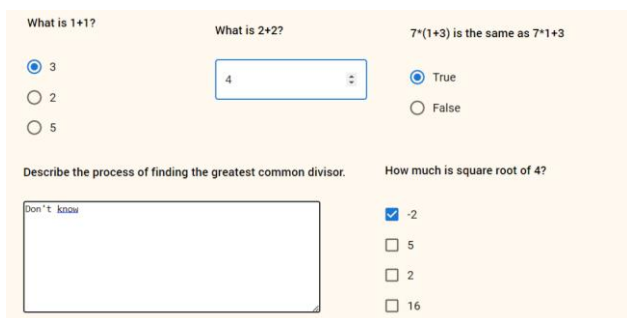
Na prvoj stranici, korisniku se nudi mogućnost izbora nekog od postojećih fajlova sa opisom procene ili unosa pitanje do njegovog fajla (slika 2). Nakon izbora, klikom na dugme "START" prelazi se na sledeći ekran.



Slika 2. Prva stranica aplikacije

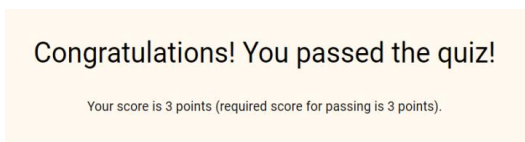
7. STUDIJA SLUČAJA

Posmatramo gramatikom opisani kviz pod nazivom "Math Quiz", koji se sastoji od 5 pitanja. Tipovi pitanja su: pitanje sa jednim mogućim odgovorom, pitanje otvorenog tipa, pitanje sa brojem kao odgovor, pitanje sa više mogućih tačnih odgovora i tačno/netačno pitanje. Osim ovoga, definisano je da kviz zahteva unos ličnih informacija, da ima ograničeno vreme trajanja od 1 sat, 4 minute i 5 sekundi, da dozvoljava preskakanje na kraj, ima definisane podrazumevane pozitivne i negativne poene od 2 i 0.5 i zahteva 3 poena da bi se smatrao položenim. U ovom slučaju, pitanje koja nema eksplicitno definisane poene, vrednovaće se podrazumevanim. Takođe, pretposlednje pitanje prihvata delimične tačne odgovore koje shodno tome buduće. Posmatraće se slučaj sa sledećim odgovorima: "3", "4", "Don't know", "-2", "true" (slika 3).



Slika 3. Pitanja i odgovori posmatranog kviza

Bodovanje bi, po pitanjima, izgledalo ovako: -0.5, 3, 0, 1, -0.5, što bi rezultiralo rezultatu od 3. Kako je 3 zahtevan broj bodova za prolazak, kviz bi se smatrao uspešno završenim (slika 4).



Slika 4. Rezultat posmatranog kviza

8. ZAKLJUČAK

Posmatrajući i analizirajući postojeća rešenja i njihove dobre i loše strane, kreiran je JSD sa ciljem da se omogući da se sa što manje koda opiše što širi opseg tipova procena. Svojom konciznošću i jednostavnom sintaksom, jezik je pogodan za upotrebu osobama iz raznih domena, bez prethodnog programerskog znanja. Osim toga, kreiranim jednostavnim korisničkim interfejsom

omogućeno je lako učitavanje opisane procene i prolazak kroz nju. Iako se težilo ka što kvalitetnijem, primenljivijem i sveobuhvatnijem jeziku i rešenju generalno, uvek postoji prostora za unapređenje. Neki od mogućih koraka u daljem razvoju mogu biti:

- Podrška za veći broj tj. širi opseg tipova procena, a i formi izvan ovog okvira (kursevi, treninzi, forme itd.).
- Omogućiti dodatne tipove pitanja kao što su: pitanja sa praznim prostorom za dopunjavanje rečenice, pitanja sa biranjem datuma kao odgovor, uslovna pitanja, pitanja za koja je potrebno priložiti sliku ili neki drugi tip fajla itd.
- Izolovati *frontend* rešenje tako da se procena može koristiti i prosleđivati dalje ciljnoj grupi zarad učestvovanja u istoj.
- Implementirati registraciju tj. prijavu.
- Integrirati rešenje sa nekim softverom za prikupljanje i analizu podataka.
- Nadograditi rešenje tako da se odgovori tj. rezultati šalju nazad serveru koji ih skladišti i dalje analizira – slično prethodnom unapređenju, s tim što bi ovo rešenje bilo "in house", te omogućilo veću slobodu i fleksibilnost.

9. LITERATURA

- [1] A. Bhat, „Surveys: What They Are, Characteristics & Examples”, <https://www.questionpro.com/blog/surveys/>, (poslednji pristup: maj 2023.)
- [2] Luciane C. Araujo, Marco A. Casanova, „[SLang: A Domain-specific Language for Survey Questionnaires](#)”, In Proceedings of the 22nd International Conference on Enterprise Information Systems, pp. 4-10, 2020. DOI: 10.5220/0009407401330144
- [3] „LiaScript A Domain-Specific Language for Interactive Online Courses”, <https://eric.ed.gov/?id=ED621589>, Association for Development of the Information Society, 2019. (poslednji pristup: maj 2023.)
- [4] F. Tomassetti, „The complete guide to (external) Domain Specific Languages”, <https://tomassetti.me/domain-specific-languages/>, 2017. (poslednji pristup: maj 2023.)
- [5] I. Dejanović, R. Vaderna, G. Milosavljević, Ž. Vuković, „TextX: A Python tool for Domain-Specific Languages implementation”, Knowledge-Based Systems, vol. 115, pp. 1-4, 2017. DOI: 10.1016/j.knosys.2016.10.023
- [6] I. Dejanović, textX, <https://www.igordejanovic.net/courses/tech/textX/2023>. (poslednji pristup: maj 2023.)

Kratka biografija:



Nevena Alavanja rođena je u Novom Sadu 1998. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – računarstvo i automatika, odbranila je 2023. god.

kontakt: nevenaalavanja33@gmail.com