

3. Dizajn strukture i ponašanje programskog proizvoda

Sažetak

Ovom lekcijom obradit će se osnove objektno orijentiranog modeliranja korištenjem nekoliko vrsta UML dijagrama koji se koriste u različitim fazama procesa razvoja softvera:

- dijagram slučajeva korištenja,
- dijagram slijeda,
- dijagram klasa, i
- dijagram aktivnosti.

Pojasnit će se sintaksa (*označavanje*¹ jezičnih elemenata), semantika (značenje jezičnih elemenata) i pragmatika (korištenje jezičnih elemenata) navedenih UML dijagrama.

Uvod

Tokom razvoja softvera dolazi do brojnih poteškoća koje se protežu izvan granica same implementacije. Budući da su u različite faze razvoja softvera obično uključene osobe s različitim znanjima i iskustvima, za učinkovitu komunikaciju potreban im je zajednički jezik. U tu svrhu koriste se jezici za modeliranje, koji služe za izradu nacрта za softverske sustave te kao osnova za implementaciju ili čak automatsko generiranje koda. U području modeliranja objektno orijentiranog razvoja softvera prevladava UML (engl. *Unified Modeling Language*). UML se pojavio 90-ih godina prošlog stoljeća kao jedan od uspješnijih pokušaja uvođenja standarda u područje modeliranja programskog inženjerstva. Moglo bi se reći da se njime opisuje programski proizvod slično kao što se CAD-om opisuje projekt kuće ili zgrade.

UML općenito razlikuje dvije vrste nacрта: dijagrame ponašanja i strukturne dijagrame. Dijagrami ponašanja pokazuju dinamiku različitih dijelova u sustavu, odnosno tok rada sustava. Ponašanje se može odrediti kroz radnje jednog objekta ili proizaći iz interakcije između više objekata. Neki od poznatijih dijagrama ponašanja su *dijagram slučajeva korištenja*, *dijagram aktivnosti* i *dijagram slijeda*. **Dijagram slučajeva korištenja** opisuje koji korisnici koriste koje funkcionalnosti sustava, no ne ulazi u specifične detalje implementacije. **Dijagram aktivnosti** opisuje mehanizme za određivanje redoslijeda kontrolnih i objektnih aktivnosti među akcijama, te je veću primjenu našao u modeliranju poslovnih

¹ notacija

procesa. **Dijagram slijeda** kao jedan od najkorištenijih UML dijagrama prikazuje interakciju sudionika i različitih dijelova sustava uzimajući u obzir vremensku komponentu.

Dijagrami strukture su po prirodi namijenjeni tome da prikažu sliku sustava ne uzimajući u obzir kretanje poruka niti vremensku komponentu u izvršavanju sustava. Svrha ove vrste dijagrama je pokazati dijelove sustava (njihovu internu strukturu) i veze koje između njih postoje. Zato se može reći da su po prirodi PI analogija tlocrtima, planovima elektroinstalacija ili vodoinstalacija za neku zgradu. Najpoznatiji dijagram strukture je *dijagram klasa*, kojim se prikazuju objekti i klase potrebni za realizaciju funkcionalnosti softverskog sustava.

Za izradu i uređivanje UML dijagrama (modela) koriste se UML alati, koji najčešće nude i dodatne funkcionalnosti, kao što su *Forward Engineering*², *Reverse Engineering*³ ili *Round-Trip Engineering*⁴. Danas postoje brojni UML alati⁵ koji se razlikuju po svojim mogućnostima, programskom jeziku i stupnju integriranosti sa programskim alatima. Neki od njih su *Visual Paradigm*, *Microsoft Visio*, *Modelio*, *Draw.IO*, *Lucidchart*... U sklopu laboratorijskih vježbi koristit ćemo alat *Visual Paradigm*, čije se besplatno *community*⁶ izdanje može preuzeti s adrese:

<http://www.visual-paradigm.com/download>.

² generiranje dijelova koda u odabranom programskom jeziku na temelju izrađenog UML modela

³ generiranje UML modela na temelju upisanog koda

⁴ neprestana sinkronizacija UML modela i programskog koda

⁵ https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools, 4.3.2022.

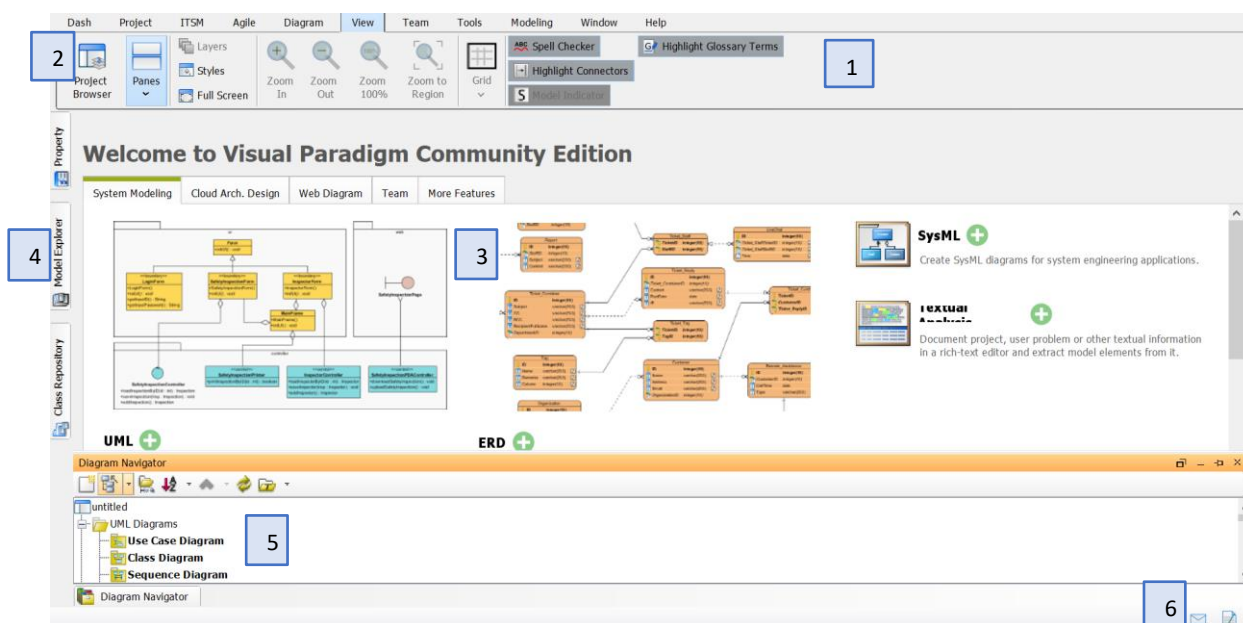
⁶ zajedničko

3.1. Visual Paradigm

Visual Paradigm je jedan od vodećih komercijalnih UML alata. U besplatnom *community* izdanju dostupne su njegove osnovne funkcionalnosti, koje u potpunosti podržavaju izradu i uređivanje svih dijagrama iz standarda UML 2⁷. U nastavku se nalazi opis rada u alatu *Visual Paradigm*.

Glavni prozor alata sastoji od sljedećih dijelova (Slika 1):

- 1) izbornik i alatne trake,
- 2) preglednik projekata,
- 3) radni prostor (ovdje je obično prikazan jedan dijagram),
- 4) razni pomoćni prikazi,
- 5) navigacija,
- 6) poruke (otvaraju se pritiskom na ikonu pisma).

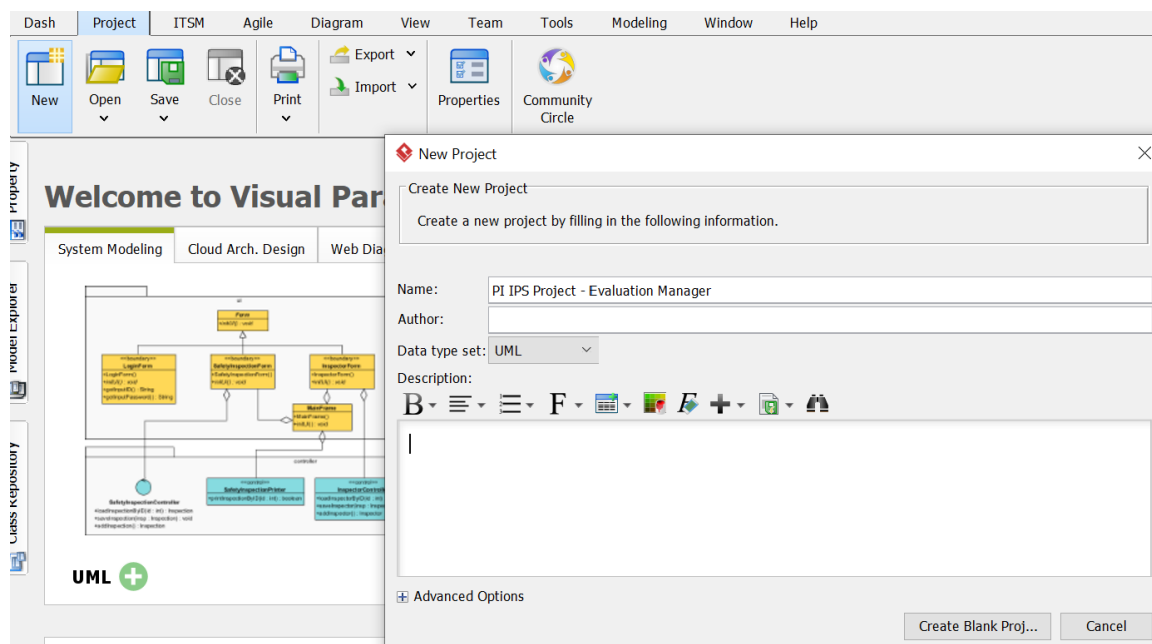


Slika 1 Visual Paradigm - glavni prozor

S obzirom da *Visual Paradigm* pohranjuje dijagrame u trenutno otvoreni projekt, najprije je potrebno stvoriti novi ili otvoriti postojeći projekt (Slika 2). Projektu je potrebno dati naziv te, ukoliko je potrebno, iz padajućeg izbornika odabrati UML kao tip podataka. Projekt se na disku pohranjuje u obliku datoteke sa .vpp⁸ ekstenzijom.

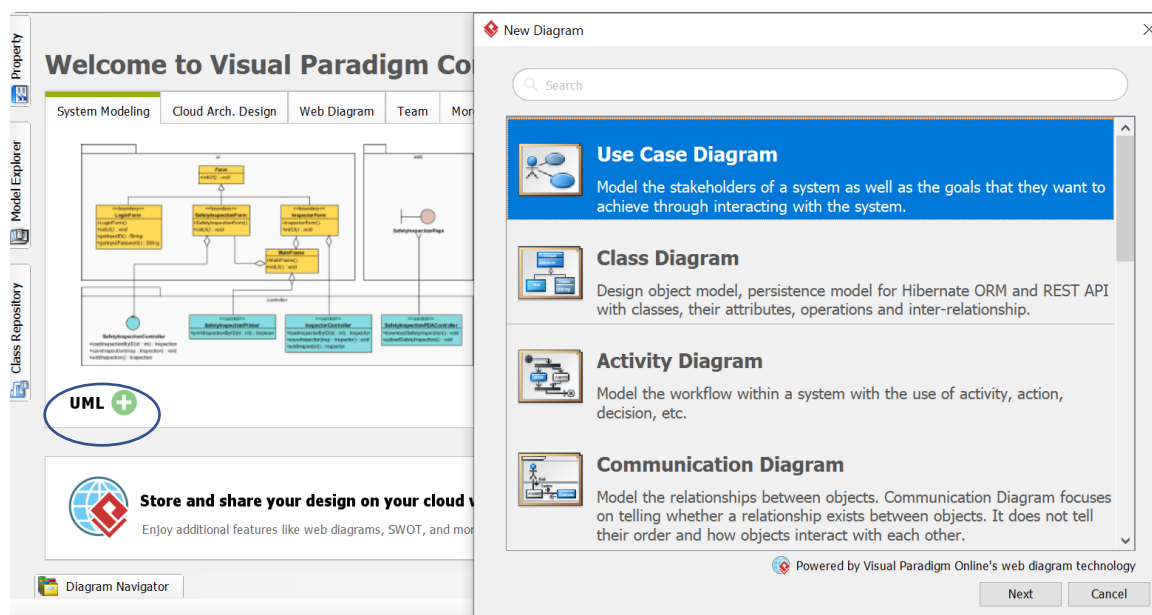
⁷ <https://www.omg.org/spec/UML/2.5.1/PDF>, str. 685, 7.3.2022.

⁸ *Visual Paradigm Project*



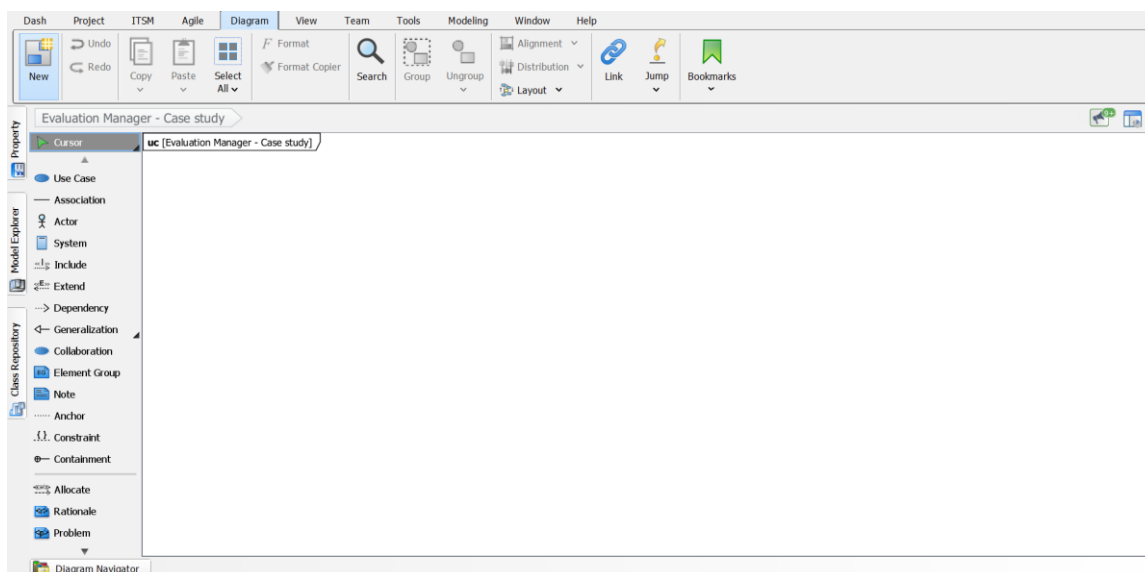
Slika 2 Otvaranje novog projekta

Zatim je potrebno odabrati jedan od ponuđenih UML dijagrama iz izbornika koji se otvara odabirom opcije **UML +** u radnom prostoru alata (Slika 3).



Slika 3 Odabir UML dijagrama

Nakon toga je potrebno odabrati način izrade dijagrama (samostalna izrada ili neki od ponuđenih predložaka) i dati naziv dijagramu. Izvršavanjem navedenih koraka otvara se novi prozor za izradu UML dijagrama (Slika 4).



Slika 4 Prozor za izradu UML dijagrama

Na vrhu prozora nalazi se alatna traka koja sadrži gumbe i alate kategorizirane u nekoliko funkcijskih izbornika. U nastavku će ukratko biti opisane najčešće korištene funkcije ovih izbornika. Primjerice, izbornik *Project* omogućuje nam otvaranje postojećeg projekta, kreiranje novog projekta, spremanje projekta i zatvaranje projekta. Izbornik *Diagram* nudi mogućnost kreiranja novog dijagrama, poništavanja promjena, kopiranja i lijepljenja dijagrama. U izborniku *View* moguće je pretraživati i pristupati projektima, otvarati prozore za pristup različitim tipovima projekata, uključiti ili isključiti *Grid*, te povećati ili smanjiti prikaz dijagrama. Izbornik *Window* nudi mogućnost otvaranja novog prozora aplikacije, upravljanja otvorenim prozorima, pristupa početnoj stranici aplikacije, konfiguriranja opcija otvorenog projekta, te integraciju s *IDE*⁹ (pr. *Eclipse*, *NetBeans*, *IntelliJ IDEA* i *Visual Studio*).

S alatne trake koja se nalazi na lijevoj strani prozora, a čiji se sadržaj prilagođava ovisno o vrsti odabranog UML dijagrama, bira se željeni grafički element. Odabrani element se zatim pomoću miša smješta u radni prostor, smanjuje, povećava i sl. Mogućnosti alata bit će detaljnije predstavljene kroz izradu različitih vrsta UML dijagrama na primjeru softvera za evidentiranje rezultata provedbe kontinuiranog praćenja (engl. *Evaluation Manager*) na kolegiju Programsko inženjerstvo. Prvi UML dijagram koji ćemo obraditi je dijagram slučaja korištenja (eng. *Use Case Diagram*).



Napomena

Visual Paradigm razumije sintaksu dijagrama, te prilikom izrade dijagrama može korisniku sugerirati ili zabraniti određene poteze (ukoliko nisu ispravni).

⁹ *Integrated Development Environment*

3.2. Dijagram slučajeve korištenja

Dijagrami slučajeve korištenja koriste se za opis mogućih načina korištenja (slučajeve korištenja) softverskog sustava. Opisuju što bi sustav trebao raditi, no ne ulaze u detalje realizacije poput strukture podataka, korištenih metoda i sl. Također opisuju koji korisnik sustava koristi koju funkcionalnost, odnosno tko će zaista koristiti sustav koji se razvija. Slučajevi korištenja predstavljaju ono što korisnik želi da sustav radi, odnosno korisničke zahtjeve.

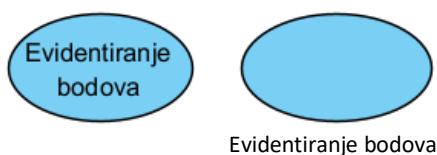
Dijagrami slučajeve korištenja daju odgovor na sljedeća pitanja :

- Što se opisuje?
- Tko stupa u interakciju sa sustavom?
- Što sudionici mogu učiniti?

3.2.1. Elementi dijagrama slučajeve korištenja

Slučaj korištenja

Slučaj korištenja predstavlja apstraktni zadatak jednog ili više sudionika koji stupaju u interakciju sa sustavom, te se njime opisuje i dokumentira funkcionalnost koju bi sustav trebao pružati. Okidač slučaja korištenja može biti sudionik (pr. nastavnik) ili događaj (pr. završetkom kontinuiranog praćenja izračunava se ukupan broj bodova koje je student prikupio tijekom semestra). Slučaj korištenja obično se predstavlja elipsom, dok se naziv slučaja korištenja upisuje unutar ili ispod elipse (Slika 5).

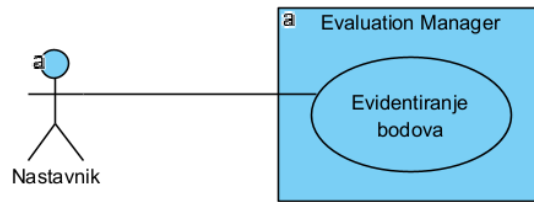


Slika 5 Notacija slučaja korištenja

Skup svih slučajeve korištenja predstavlja funkcionalnost koju sustav treba pružiti. Pritom je bitno imati na umu da se mnogi mali slučajeve korištenja koji imaju isti cilj mogu predstaviti kao jedan slučaj korištenja (npr. u slučaj korištenja *Evidentiranje bodova* uključeni su scenariji unosa, izmjene i brisanja ocjene).

Sustav

Slučajevi korištenja obično su grupirani unutar pravokutnika, koji predstavlja granice sustava kojeg opisujemo. Na slici Slika 6 je prikazan slučaj korištenja *Evidentiranje bodova*, koji se nalazi unutar granica sustava, dok se sudionik (*Nastavnik*) nalazi izvan granica sustava.



Slika 6 Granice sustava

Upozorenje

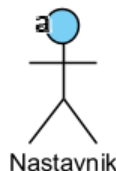
Sudionici se ne nalaze unutar sustava, već izvan granica sustava!

Sudionik

Sudionici ne predstavljaju konkretnog korisnika, nego uloge koje korisnici preuzimaju korištenjem sustava (Slika 7). Dakle, ukoliko korisnik preuzme određenu ulogu, ovlašten je za izvršavanje slučaja korištenja povezanog s tom ulogom.

Tipovi sudionika su:

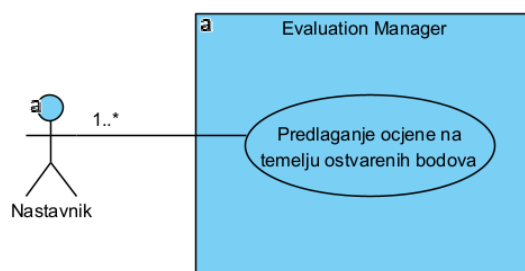
- aktivni (inicira realizaciju slučaja korištenja) ili pasivni (pruža funkcionalnost za realizaciju slučaja korištenja),
- ljudski (pr. nastavnik) ili neljudski (pr. server),
- primarni (ostvaruje direktnu dobit) ili sekundarni (ne ostvaruje direktnu dobit).



Slika 7 Notacija sudionika

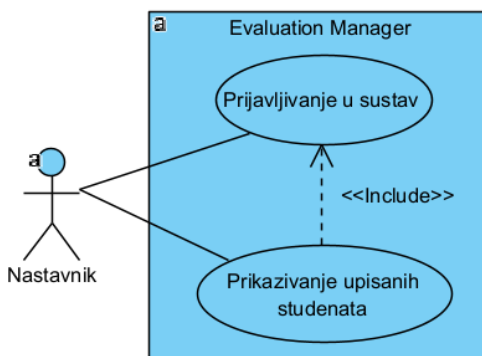
Veze

Sudionici su povezani sa slučajevima korištenja punom linijom (asocijacijom) koja označava da sudionik stupa u interakciju sa sustavom i koristi određenu funkcionalnost. Svaki sudionik mora biti povezan s najmanje jednim slučajem korištenja, i obrnuto. Veza je uvijek binarna, što znači da je definirana između jednog korisnika i jednog slučaja korištenja. Na krajevima veze moguće je odrediti brojnost. Ukoliko se na kraju veze sa sudionikove strane nalazi broj veći od 1, to znači da je u realizaciju slučaja uključeno više od jedne instance sudionika (Slika 8). Veza na strani slučaja korištenja uglavnom se ne ograničava, te ju u većini slučajeva nije potrebno definirati. Ukoliko se na krajevima veze ne nalazi nikakva vrijednost, podrazumijeva se da je zadana vrijednost 1.

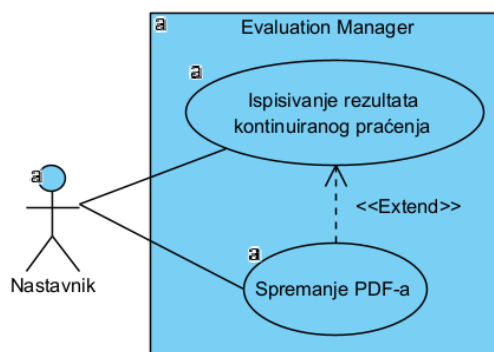


Slika 8 Brojnost na kraju veze sa sudionikove strane

Osim veza između sudionika i slučajeva korištenja, na dijagramu slučajeva korištenja moguće je definirati i veze između slučajeva korištenja. Razlikujemo `<<Uključi>>` (engl. `<<include>>`) i `<<Proširi>>` (engl. `<<extend>>`) vezu. `<<Uključi>>` veza označava da osnovni slučaj korištenja sadrži ponašanje uključenog slučaja korištenja. Osnovni slučaj korištenja nije kompletan bez uključenog slučaja korištenja, te traži izvršavanje uključenog slučaja korištenja prije izvršavanja vlastite funkcionalnosti. S druge strane, uključen slučaj korištenja može se izvršiti samostalno ako je povezan na minimalno jednog sudionika. U primjeru na slici Slika 9, *Prikazivanje upisanih studenata* predstavlja osnovni slučaj korištenja, dok *Prijavljivanje u sustav* predstavlja uključen slučaj korištenja. U konkretnom primjeru `<<include>>` vezom želimo prikazati da nastavnik ne može pristupiti evidenciji studenata ukoliko nije prijavljen u sustav, odnosno ukoliko mu je isteklo vremensko ograničenje sesije. Drugim riječima, svaki puta kada nastavnik želi pristupiti evidenciji studenata, u pozadini aplikacije provjerit će se status nastavnikove prijave u aplikaciju.

Slika 9 `<<Uključi>>` veza

`<<Proširi>>` veza označava da se ponašanje osnovnog slučaja korištenja može proširiti ponašanjem dodatnog slučaja korištenja, ali i ne mora. Oba slučaja korištenja mogu se izvršiti neovisno jedan o drugom ako su neovisno povezani na minimalno jednog sudionika. Tijekom provođenja osnovnog slučaja korištenja, sudionik odlučuje hoće li i prošireni slučaj korištenja biti izvršen. U primjeru na slici Slika 10 *Ispisivanje rezultata kontinuiranog praćenja* predstavlja osnovni slučaj korištenja, dok slučaj korištenja *Spremanje PDF-a* predstavlja uključen slučaj korištenja.



Slika 10 <<Proširi>> veza

3.2.2 Primjer izrade dijagrama slučajeve korištenja u alatu Visual Paradigm



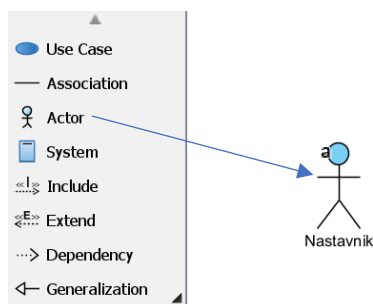
Ako dosad niste, kreirajte novi projekt sukladno uputama iz poglavlja 3.1.

Izrada dijagrama slučajeve korištenja u alatu *Visual Paradigm* biti će demonstrirana na primjeru softvera za evidentiranje rezultata provedbe kontinuiranog praćenja (engl. *Evaluation Manager*) na kolegiju Programsko inženjerstvo. Da bismo izradili dijagram, u alatu *Visual Paradigm* trebamo kreirati novi projekt i imenovati ga prema

uputama iz poglavlja 3.1. Za izradu dijagrama slučajeve korištenja potrebno je odabrati *Use Case Diagram*. Prije izrade dijagrama potrebno je identificirati sudionike i slučajeve korištenja. Kao podloga za izradu ovih dijagrama koristit će se funkcionalni zahtjevi s najvećim prioritetom definirani u poglavlju *Specifikacija softverskih zahtjeva*:

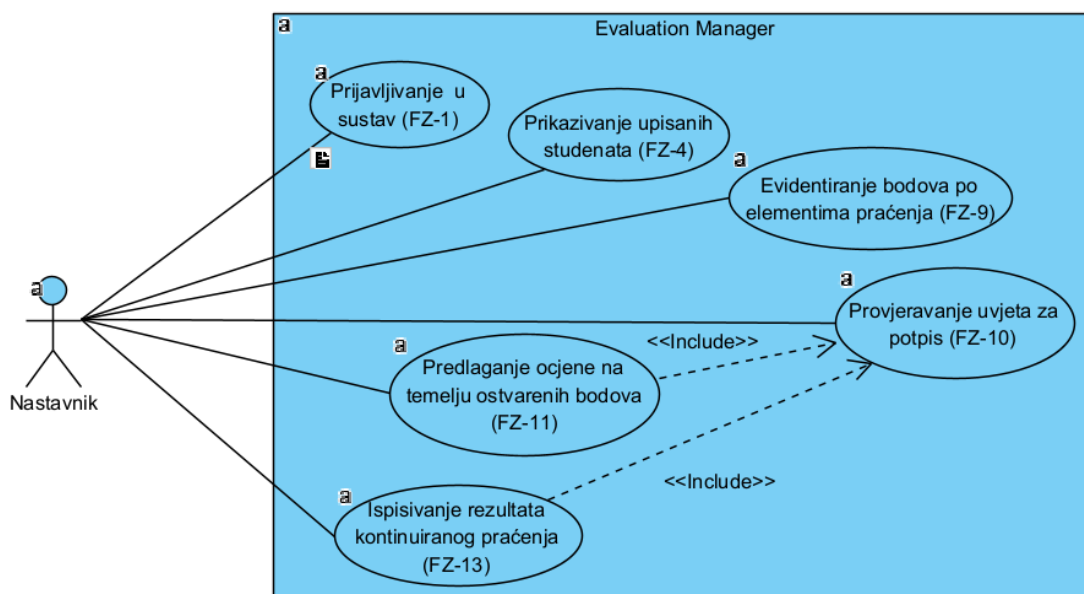
- 1) Sustav će omogućiti pristup samo autentificiranim korisnicima (FZ-1).
- 2) Sustav će omogućiti prikaz studenata upisanih na kolegij (FZ-4).
- 3) Sustav će omogućiti evidentiranje bodova ostvarenih prilikom evaluacije za svakog upisanog studenta po definiranim elementima praćenja (FZ-9).
- 4) Sustav će za svakog studenta moći odrediti jesu li ostvareni uvjeti za potpis ili ne (FZ-10).
- 5) Po završetku kontinuiranog praćenja, sustav će za studente koji su ostvarili uvjet za potpis predložiti ocjenu na temelju ukupnog broja ostvarenih bodova. (FZ-11).
- 6) Sustav će omogućiti ispis rezultata kontinuiranog praćenja za studente koji su ostvarili potpis. Pri tome za svakog studenta treba biti vidljiv broj bodova pojedinačnih elemenata praćenja, ukupan broj bodova, te prijedlog ocjene (FZ-13).

U specifikacijama je navedeno da će sudionici softvera *Evaluation Manager* biti nastavnici na kolegiju Programsko inženjerstvo. S obzirom da će svi nastavnici koristiti softver na isti način i imati istu razinu prava, potrebno je definirati samo jednu ulogu – *Nastavnik*. Da bismo nacrtali sudionika, s alatne trake koja se nalazi na lijevoj strani prozora potrebno je odabrati grafički element *Actor* (Slika 11). Odabrani element potrebno je pomoću miša smjestiti u radni prostor i dati mu naziv *Nastavnik*.



Slika 11 Kreiranje sudionika

U specifikacijama je također navedeno na koji način će softver biti korišten, te je moguće vrlo jednostavno kreirati moguće slučajeve korištenja. Slučaj korištenja crta se na identičan način kao i sudionik, no umjesto grafičkog elementa *Actor* potrebno je odabrati grafički element *Use Case*. Slučajeve korištenja potrebno je smjestiti unutar sustava (grafički element *System*). Naposljetku je između sudionika i slučaja korištenja potrebno nacrtati ispravne veze (*asocijacija*, *<<uključiti>>* ili *<<proširi>>*), koje se također nalaze u izborniku s lijeve strane prozora. Između nastavnika i svih slučajeva korištenja crtamo asocijaciju (engl. *Association*), dok između slučajeva korištenja crtamo vezu *<<uključiti>>*, koja označava da navedeni slučajevi korištenja uključuju slučaj korištenja *Provjeravanje uvjeta za potpis*. Drugim riječima, samo studenti koji imaju pravo na potpis mogu dobiti ocjenu, te će samo studenti koji imaju pravo na potpis biti prikazani na generiranom izvještaju. Dijagram slučajeva korištenja izrađen na temelju informacija iz specifikacija prikazan je na slici Slika 12.



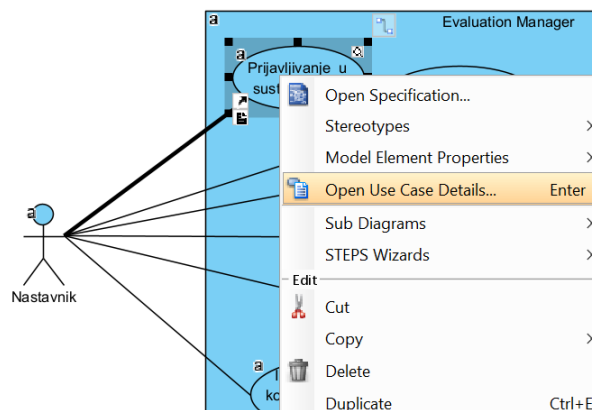
Slika 12 Dijagram slučajeva korištenja

3.2.3. Opis slučajeve korištenja

Kako bi se osiguralo da dijagrami slučajeve korištenja budu jasni, važno je odabrati kratke i jasne nazive slučajeve korištenja, te je iste potrebno jasno i nedvosmisleno opisati. Za opis slučajeve korištenja predlažu se Cockburnove¹⁰ smjernice, prema kojima strukturirani pristup opisu slučajeve korištenja treba sadržavati sljedeće informacije:

- Naziv
- Kratki opis
- Sudionici
- Okidač (događaj koji pokreće slučaj korištenja)
- Preduvjet za uspješno izvršenje
- Stanje sustava nakon uspješnog izvršenja
- Iznimke
- Stanje sustava nakon pojave iznimke
- Standardni proces (pojedinačni koraci koje treba poduzeti)
- Alternativni procesi (odstupanja od standardnog procesa)

Visual Paradigm community verzija **djelomično podržava** unos detalja vezanih uz izvršavanje slučajeve korištenja. Za pristup navedenim mogućnostima potrebno je odabrati slučaj korištenja, u padajućem izborniku odabrati opciju *Open Use Case Details* (Slika 13), te pronaći kartice *Info* i *Details*.

















Slika 13 Otvaranje opisa slučaja korištenja

S obzirom da su mogućnosti *community* verzije ovog alata relativno ograničene, u nastavku rada je na primjeru slučaja korištenja *Prijavljivanje u sustav (FZ-1)* pojašnjeno na koji način se ranije navedene smjernice mogu primijeniti unutar dostupnih opcija alata.

¹⁰ Cockburn, Alistair. Writing effective use cases. Pearson Education India, 2001.

Veći dio detalja vezanih uz slučaj korištenja upisuje se na kartici *Info*, u polju *Description*: kratki opis, okidači, iznimke, stanje sustava nakon pojave iznimke, standardni proces i alternativni procesi (Slika 14).

Prijavljivanje u sustav (FZ-1)

Info	Use Case Notes	Flow of Events	Details	Requirements	Diagrams	Test Plan	References
Rank:	High  						
ID:	FZ-1						
Status:	Identify  Next						
Justification:							
Primary Actors:	Nastavnik						
Supporting Actors:							
Task Pool:	Link to...						
Description:	<div>            </div> <p>Kratki opis: Prije korištenja aplikacije, od svakog se nastavnika traži prijava u aplikaciju kako bi se ograničio pristup osjetljivim podacima studenata, te kako bi se moglo razlikovati koje rezultate je evidentirao svaki od nastavnika.</p> <p>Okidači: Nastavnik traži pristup aplikaciji.</p> <p>Iznimke: Ne postoji nastavnik s upisanim korisničkim podacima.</p> <p>Stanje sustava nakon pojave iznimke: Nastavnik nije prijavljen u aplikaciju.</p> <p>Standardni proces: (1) nastavnik unosi korisničke podatke, (2) nastavnik traži pristup aplikaciji, (3) sustav dohvaća unesene podatke, (4) sustav provjerava ispravnost unesenih korisničkih podataka, (5) sustav šalje poruku o uspješnosti prijave, (6) nastavnik pristupa evidenciji.</p> <p>Alternativni procesi: (1-4) kao i standardni proces, (5') aplikacija šalje poruku o grešci, (6') nastavnik prima poruku o grešci.</p>						
<input type="checkbox"/> Abstract <input type="checkbox"/> Leaf <input type="checkbox"/> Root							

Slika 14 Opis slučaja korištenja - kartica *Info*

Manji dio detalja vezanih uz slučaj korištenja upisuje se na kartici *Details*: preduvjet (engl. *Preconditions*) i stanje sustava nakon uspješnog izvršenja (engl. *Post-conditions*) (Slika 15).

Prijavljivanje u sustav (FZ-1)

Info	Use Case Notes	Flow of Events	Details	Requirements	Diagrams	Test Plan	References
Level:	User						
Complexity:	Low						
Use Case Status:	Name Only						
Implementation Status:							
<input checked="" type="radio"/> Preconditions:	Nastavnik posjeduje korisničke podatke.						
<input type="radio"/> Post-conditions:	Nastavnik je prijavljen u aplikaciju.						
Author:							
<input type="radio"/> Assumptions:							

Slika 15 Opis slučaja korištenja - kartica *Details*

Iz opisa slučaja korištenja izlazimo odabirom jedne od opcija sa slike Slika 16, koje se nalaze u gornjem desnom kutu prozora. Sve tri opcije nude mogućnost povratka na jedan od dijagrama slučajeva korištenja koji se nalazi unutar trenutno otvorenog projekta.



Slika 16 Izlaz iz opisa slučaja korištenja

Nakon izrade opisa slučaja korištenja, na dijagramu slučajeva korištenja će uz opisani slučaj korištenja biti prikaza ikona izvještaja. Ukoliko kliknemo na tu ikonu, otvorit će se detalji slučaja korištenja koje smo ranije upisali.

Budući da *community* verzija alata **ne omogućuje** izvoz opisa slučajeva korištenja u obliku teksta, preostali slučajevi korištenja će u svrhu lakše manipulacije podacima biti opisani unutar tablica. Primijetit ćete da neki slučajevi korištenja nemaju definirane iznimke, stanje sustava nakon pojave iznimke ili alternativni proces. To je sasvim u redu, budući da se radi o vrlo jednostavnim slučajevima korištenja kod kojih se navedena stanja ne javljaju.

Opis slučaja korištenja Prikazivanje upisanih studenata (FZ-4)

Naziv	Opis
Naziv	Prikazivanje upisanih studenata
Kratki opis	Nastavnici moraju u svakom trenutku moći vidjeti studente koji su upisani u kolegij kako bi mogli evidentirati njihove aktivnosti.
Sudionici	Nastavnik
Okidači	Nastavnik pristupa evidenciji studenata.
Preduvjet	Nastavnik je prijavljen u aplikaciju.
Stanje sustava nakon uspješnog izvršenja	Nastavnik ima uvid u evidenciju studenata koji su upisani u kolegij.
Iznimke	Nije primjenjivo.
Stanje sustava nakon pojave iznimke	Nije primjenjivo.
Standardni proces	(1) Nastavnik traži prikaz studenata upisanih na kolegij. (2) Sustav prikazuje studente upisane na kolegij.
Alternativni procesi	Nije primjenjivo.

Opis slučaja korištenja Evidentiranje bodova po elementima praćenja (FZ-9)

Naziv	Opis
Naziv	Evidentiranje bodova po elementima praćenja.
Kratki opis	Nastavnici upisuju bodove studenata u skladu s definiranim elementima praćenja.
Sudionici	Nastavnik
Okidači	Nastavnik pristupa evaluaciji.
Preduvjet	Nastavnik je prijavljen u aplikaciju. Studenti su evidentirani u sustav.
Stanje sustava nakon uspješnog izvršenja	Evidentirani su ostvareni bodovi studenata.
Iznimke	Nije odabran nijedan student. Uneseni broj bodova veći je od maksimalnog broja bodova za određenu aktivnost. Unos negativnog broja bodova.
Stanje sustava nakon pojave iznimke	Ostvareni bodovi nisu evidentirani.
Standardni proces	(1) Nastavnik pristupa popisu studenata upisanih u kolegij. (2) Sustav prikazuje popis studenata upisanih u kolegij. (3) Nastavnik bira studenta kojem želi evidentirati bodove. (4) Sustav sprema studenta kojem je potrebno evidentirati bodove. (5) Nastavnik otvara evaluacijsku formu. (6) Sustav prikazuje evaluacijsku formu. (7) Nastavnik bira element praćenja za koji je potrebno evidentirati bodove. (8) Sustav prikazuje opcije za odabrani element praćenja. (9) Nastavnik unosi broj bodova. (10) Sustav prikazuje unesene bodove. (11) Nastavnik traži pohranu podataka o evaluaciji. (12) Sustav pohranjuje podatke o evaluaciji. (13) Sustav zatvara evaluacijsku formu.
Alternativni procesi	A (1 - 8) Kao i standardni proces. (9') Nastavnik mijenja ili briše podatke o broju bodova. (10-13) Kao i standardni proces. B (1-10) Kao i standardni proces. (11') Nastavnik odustaje od spremanja promjena. (12' - 13') Sustav zatvara evaluacijsku formu bez spremanja promjena. C (1 - 8) Kao i standardni proces. (9') Nastavnik unosi neispravan broj bodova (negativan broj ili broj koji je veći od maksimalnog broja bodova za odabranu aktivnost). (10') Sustav prikazuje poruku pogreške. (9-13) Kao i standardni proces (pod uvjetom pravilnog unosa broja bodova). D (1-2) Kao i standardni proces. (3') Nastavnik otvara evaluacijsku formu bez odabira studenta. (4') Sustav prikazuje poruku pogreške. (3-13) Kao i standardni proces (pod uvjetom odabira studenta).

Opis slučaja korištenja Provjeravanje uvjeta za potpis (FZ-10)

Naziv	Opis
Naziv	Provjeravanje uvjeta za potpis.
Kratki opis	Za svakog je studenta potrebno odrediti jesu li ostvareni uvjeti za potpis ili ne. Kako bi ostvario uvjete za potpis, student mora imati evidentirane barem minimalne bodove iz svih modelom definiranih elemenata praćenja. Nastavnik dobiva uvid u popis studenata koji su ostvarili pravo na potpis putem izvještaja s rezultatima kontinuiranog praćenja.
Sudionici	Nastavnik
Okidači	Nastavnik traži uvid u rezultate studenata koji su ostvarili pravo na potpis.
Preduvjet	Nastavnik je prijavljen u aplikaciju. Studenti su evidentirani u sustav. Studentima su dodijeljeni bodovi za sve elemente praćenja. Zbrojeni su bodovi studenata.
Stanje sustava nakon uspješnog izvršenja	Utvrđeno je koji studenti imaju pravo na potpis..
Iznimke	Nije primjenjivo.
Stanje sustava nakon pojave iznimke	Nije primjenjivo.
Standardni proces	(1) Nastavnik traži uvid u rezultate studenata koji su ostvarili pravo na potpis. (2) Sustav dohvaća podatke o evaluaciji studenata. (3) Sustav provjerava jesu li bodovani svi elementi praćenja. (4) Sustav izračunava broj bodova za studente s kompletnom evaluacijom. (5) Sustav utvrđuje je li student ostvario pravo na potpis. (6) Sustav sprema rezultate studenata koji su ostvarili pravo na potpis. (7) Sustav prikazuje izvještaj s rezultatima studenata koji su ostvarili pravo na potpis.
Alternativni procesi	Nije primjenjivo.

Opis slučaja korištenja Predlaganje ocjene na temelju ostvarenih bodova (FZ-11)

Naziv	Opis
Naziv	Predlaganje ocjene na temelju ostvarenih bodova.
Kratki opis	Po završetku kontinuiranog praćenja, sustav će za studente koji su ostvarili uvjet za potpis predložiti ocjenu na temelju ukupnog broja ostvarenih bodova.
Sudionici	Nastavnik
Okidači	Nastavnik traži uvid u rezultate studenata koji su ostvarili pravo na ocjenu.
Preduvjet	Nastavnik je prijavljen u aplikaciju. Studenti su evidentirani u sustav. Studentima su dodijeljeni bodovi za sve elemente praćenja. Provjereni su uvjeti za potpis studenata.
Stanje sustava nakon uspješnog izvršenja	Evidentirane su ocjene studenata.
Iznimke	Nije primjenjivo.
Stanje sustava nakon pojave iznimke	Nije primjenjivo.

Standardni proces	(1) Nastavnik traži uvid u rezultate studenata koji su ostvarili pravo na ocjenu. (2) Sustav dohvaća podatke o evaluaciji studenata. (3) Sustav provjerava je li student ostvario pravo na potpis. (4) Sustav provjerava je li ukupan broj bodova dovoljan za ocjenu. (5) Sustav uspoređuje ukupan broj bodova studenata koji imaju dovoljno bodova za ocjenu s definiranom bodovnom skalom i izračunava ocjene studenata. (6) Sustav sprema ocjene studenata. (7) Sustav prikazuje izvještaj s rezultatima studenata koji su ostvarili pravo na ocjenu.
Alternativni procesi	Nije primjenjivo.

Opis slučaja korištenja Ispisivanje rezultata kontinuiranog praćenja (FZ-13)

Naziv	Opis
Naziv	Ispisivanje rezultata kontinuiranog praćenja
Kratki opis	Sustav će izraditi izvještaj sa popisom studenata koji nisu ostvarili pravo na potpis.
Sudionici	Nastavnik
Okidači	Nastavnik pristupa generiranju izvještaja.
Preduvjet	Nastavnik je prijavljen u aplikaciju. Studenti su evidentirani u sustav. Studentima su dodijeljeni bodovi. Provjereni su uvjeti za potpis studenata. Studentima je evidentirana ocjena.
Stanje sustava nakon uspješnog izvršenja	Generiran je izvještaj s rezultatima kontinuiranog praćenja.
Iznimke	Nije primjenjivo.
Stanje sustava nakon pojave iznimke	Nije primjenjivo.
Standardni proces	(1) Nastavnik traži generiranje izvještaja. (2) Sustav pokreće generiranje izvještaja. (3) Sustav dohvaća rezultate kontinuiranog praćenja. (4) Sustav prikazuje rezultate kontinuiranog praćenja. (5) Nastavnik traži zatvaranje izvještaja. (6) Sustav zatvara izvještaj.
Alternativni procesi	Nije primjenjivo.

Pitanja

1. Što je UML?
2. Koje vrste nacrtu razlikuje UML? Pojasnite razliku između njih.
3. Čemu služe dijagrami slučajeve korištenja?
4. Navedite tipove sudionika.
5. Objasnite razliku između <<uključuje>> i <<proširuje>> veze.
6. Na koji način je moguće identificirati sudionike?
7. Na koji način je moguće identificirati slučajeve korištenja?
8. Koje informacije treba sadržavati detaljni opis slučajeve korištenja?