



SOFTVERSKO INŽENJERSTVO

školska 2024/2025 godina

Vežba 6: Primena AI alata za modelovanje

Kao što već znamo, **UML (Unified Modeling Language)** je **standardizovani jezik** za:

- Modelovanje strukture (npr. klase, objekti)
- Modelovanje ponašanja (npr. tokovi aktivnosti, poruke između objekata)

Ove dijagrame možemo primeniti u:

- Analizi zahteva
- Dizajnu softvera
- Dokumentaciji projekta
- Komunikaciji sa timom ili klijentima

Zašto izabrati AI?

- **Tradicionalno** pravljenje dijagrama je ručno, sporo, lako se prave greške.
- **Pomoću AI** možemo automatski generisati nacрте, ubrzati razvoj i fokusirati se na ispravnost sistema. Naravno, treba i ovde voditi računa da se ne pojave neke nelogičnosti, stoga je i dalje preporučeni pristup **AI alati + manuelna intervencija**.

👉 **Prompt je ključ** > Što je detaljniji opis sistema, to je bolji UML!

Dobar prompt treba da sadrži:

- Ko su glavni akteri (korisnici)
- Šta glavni akteri rade
- Koje glavne klase postoje
- Odnose između klasa (nasleđivanje, asocijacije)
- Tok poruka kod sekvencijalnih procesa

Šablon za pisanje dobrog prompta:

- ***"Napiši mi UML klasni dijagram za sistem [naziv sistema]. U sistemu postoje sledeće klase: [klase sa atributima i metodama]. Pojasni odnose među klasama (nasleđivanje, asocijacije, kompozicije)."***

Primer za online prodavnicu:

- *"Napiši UML klasni dijagram za online prodavnicu. Postoje korisnici (username, password, login, purchase), proizvodi (name, price, description), i administratori (add product, delete product). Korisnici mogu kupovati proizvode. Admini nasleđuju korisnike."*

Primer za upravljanje bibliotekama

- *"Napiši UML klasni dijagram za sistem biblioteke. U sistemu postoje klase: Član (ime, prezime, članarina, iznajmiKnjigu()), Knjiga (naslov, autor, ISBN, status), Bibliotekar (dodajKnjigu(), ukloniKnjigu(), izdajKnjigu()). Članovi mogu iznajmiti knjige. Bibliotekari su posebna vrsta zaposlenih."*

Gde i kako generisati UML sa AI?

Evo najboljih opcija:

Alat/Sajt	Funkcija	Gde ubaciti kod	Prednosti	Mane
ChatGPT + PlantUML	Generiše PlantUML kod	PlantText.com ili lokalno	Precizno, fleksibilno, prilagodljivo	Potreban dodatni korak renderovanja
Lucidchart AI	Pravi vizuelne dijagrame direktno	Sve u browseru	Lep dizajn, lako uređivanje	Ograničenja u besplatnoj verziji
Mermaid Live Editor	Generiše Mermaid dijagrame	Mermaid Live Editor sajt	Jednostavan, odmah vidljiv dijagram	Ograničena grafička moć
Whimsical AI	Brza izrada flowchart dijagrama	Sve u aplikaciji	Super brzo, pregledno	Nije 100% UML standard

Kako funkcioniše proces?

1. Opisati problem prirodnim jezikom.
2. Postaviti upit: Zamoliti ga da generiše UML dijagram (class, use case i/ili sequence).
3. Dobijeni tekstualni kod (PlantUML, Mermaid) ubaciti u neki render alat.
4. Pregledati i ispraviti generisani dijagram po potrebi.

Klasni dijagram (Class Diagram)

Prikazuje klase u sistemu, njihove attribute, metode i odnose među klasama (nasleđivanje, asocijacije). Primer opisa koji može da se da AI-u:

"Generiši UML klasni dijagram za online prodavnicu: imamo korisnike koji mogu da kupuju proizvode, proizvodi imaju cenu i opis, administratori mogu da dodaju i brišu proizvode."

Mogući AI odgovor (PlantUML kod):

```
@startuml
class User {
    - username: String
    - password: String
    + login()
    + purchaseProduct()
}

class Product {
    - name: String
    - price: double
    - description: String
}

class Admin {
    + addProduct()
    + deleteProduct()
}

User --> Product : purchase
Admin --|> User
@enduml
```

Ovako generisan kod se može kopirati u PlantText ili neki drugi PlantUML editor da se dobije vizuelni klasni dijagram.

Use Case diagram

Prikazuje kako akteri (korisnici) komuniciraju sa sistemom kroz funkcionalnosti. Primer opisa koji može da se da AI-u:

"Napravite use case diagram za biblioteku: korisnici mogu da pozajme knjigu, vrate knjigu, i pregledaju katalog."

Mogući AI odgovor:

```
@startuml
actor User
actor Librarian

User --> (Borrow Book)
User --> (Return Book)
User --> (Browse Catalog)
Librarian --> (Manage Inventory)
@enduml
```

Sekvencijalni dijagram (Sequence Diagram)

Pokazuje redosled komunikacije između objekata u sistemu tokom izvršavanja neke funkcije. Primer opisa koji može da se da AI-u:

"Napravite sekvencijalni dijagram za proces online naručivanja: korisnik bira proizvod, dodaje ga u korpu, potvrđuje narudžbinu, i sistem šalje potvrdu."

Mogući AI odgovor:

```
@startuml
actor User
participant "Online Store" as Store
participant "Payment Gateway" as Payment

User -> Store : Browse products
User -> Store : Add product to cart
User -> Store : Confirm order
Store -> Payment : Process payment
Payment --> Store : Payment confirmation
Store --> User : Order confirmation
@enduml
```

Pregled alata za render:

1. PlantUML kod

- Otvori sajt: <https://planttext.com/> ili <https://plantuml.com/> (a odatle Online Server)
- U adekvatno tekst polje **nalepi** kod koji ti AI generiše.
- Klikni na "**Refresh**" ili "**Generate**".
- Dijagram se odmah pojavljuje.

2. Lucidchart AI

- Registruj se na <https://www.lucidchart.com/>
- Klikni **New Document > AI Diagram Generator**.
- Napiši tekstualni opis Sistema.
- Lucidchart automatski crta dijagram.

3. Mermaid kod

- Otvori sajt: <https://mermaid-js.github.io/mermaid-live-editor/>
- Nalepi Mermaid kod koji AI generiše.
- Klikni na "**Generate Diagram**".

Bonus saveti za još bolje promptove:

- Uvek naglasi KOJE funkcije (metode) imaju klase.
- Napiši odnose među klasama (npr. **nasleđivanje** ako neka klasa nasledjuje drugu, **asocijaciju** ako sarađuju).
- Ako želiš sekvencijalni ili use-case dijagram, samo u promptu dodaj "i nacrtaj use case" ili "i nacrtaj sekvencijalni za funkciju [npr. login]."

Prednosti korišćenja AI za UML:

- 🕒 Štedi **vreme**.
- 🧠 Pomaže kod **brainstorminga** kada nismo sigurni kako da modelujemo sistem.
- 🎨 Vizualizuje brzo i jednostavno i velike sisteme.
- 🎯 Idealno za **iterativno projektovanje** – lako možeš dodavati nove zahteve i AI može brzo izgenerisati izmene.