

MakeitTrue-MiT

Dokument softverske arhitekture

Verzija 1.0

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 03.01.2022 | 1.0 | Početna verzija dokumenta | Tanja Veselinović |
| 18.02.2022 | 1.0 | Dorada dokumenta | Tanja Veselinović |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Uvod 4

1.1 Svrha 4

1.2 Područje 4

1.3 Definicije, skraćenice i akronimi 4

1.4 Reference 4

1.5 Pregled 4

2. Predstavljanje arhitekture 4

3. Pogled slučajeva korišćenja 4

3.1 Realizacija slučajeva korišćenja 4

4. Arhitektonski ciljevi i ograničenja 4

5. Logički pogled 4

5.1 Slojevi 5

6. Pogled na procese 5

7. Pogled na implementaciju 5

7.1 Slojevi 5

8. Veličina i performanse 5

9. Kvalitet 5

Dokument softverske arhitekture

# Uvod

Dokument softverske arhitekture pruža osnovne informacije o arhitekturi softvera. Uključuje namjenu, obim i definicije. Arhitektura softvera predstavlja značajnu postavku u dizajnu i implementaciji softverskog proizvoda.

## Svrha

Dokument softverske arhitekture pruža sveobuhvatan arhitektonski pregled sistema, koristeći različite arhitektonske poglede za prikaz različitih aspekata sistema. Namijenjen je da prikaže značajne arhitektonske odluke koje su realizovane tokom izrade sistema.

## Područje

Dokument softverske arhitekture pruža arhitektonski pregled grafičkog editora kombinatornih mreža MiT. Grafički editor razvijaju članovi grupe 1. Proizvod je namijenjen za iscrtavanje logičkih kola.

## Definicije, skraćenice i akronimi

Sve definicije, skraćenice i akronimi se nalaze u dokumentu Rječnik.

## Reference

…

## Pregled

U nastavku dokumenta je dat pregled i opis arhitekture softvera, njenih ciljeva, i različitih pogleda na samu arhitekturu sistema.

# Predstavljanje arhitekture

U dizajnu softvera kao osnovni šabloni su korišteni arhitektonski šablon MVC (Model-View-Controller) i Observer. On podrazumjeva razdvajanje modela podataka od načina njegovog prikaza, te upravljanje i komunikaciju među njima koja se obavlja pomoću kotrolera. Ova podjela uslovljava i implementacionu podjelu na različite module koji odgovaraju pojedinim aspektima sistema. U našem slučaju je korišten modifikovana MVC arhitektura I Observer gdje je kontroler izostavljen zbog prirode softvera. Sem MVC arhitektonskog šablona korišteni su i drugi šabloni: Command, State, Strategy, Memento, Bridge Composite, Decorator...

# Pogled slučajeva korišćenja

Ovaj pogled opisuje sve značajne slučajeve korištenja sa aspekta arhitekture sistema. Detaljniji opisi slučajeva korištenja se nalaze u dokumentu Realizacija slučajeva korištenja.

# Arhitektonski ciljevi i ograničenja

Upotrebom ranije pomenutih dizajn šablona postiže se precizno razdvajanje odgovornosti i zaduženja pojedinih klasa. To kao rezultat daje kvalitetnu strukturu u kojoj je detekcija grešaka i nadogradnja olakšana. Prilikom proširivanja softvera ili dorade već postojećih komponenti, smanjuje se obim klasa u kojima se mogu otklanjati potencijalne greške i koje se moraju ponovno testirati što dovodi do znatne uštede vremena i povećanja kvaliteta softvera.

# Logički pogled

Implementacija softvera je razdvojena na programske module koji su odgovarajuće povezani. Svaki modul se sastoji od odgovarajućeg broja klasa čija saradnja realizuje slučajeve korištenja.

## Slojevi

* *Model* programski paket predstavlja dio sistema koji sadrži klase za stanje samog sistema i njihovu prezentaciju u operativnoj memoriji računara. On služi za čuvanje informacija o sistemu, trenutnim dokumentima i odgovoran je za konkretne podatke koji su potrebni radu softvera. Dizajn šabloni koji su iskorišteni za razradu modela su: Command, Memento za realizaciju operacija undo i redo, State za definisanje stanja aplikacije i dijagrama, Strategy za svojstva aplikacije, Decorator, Bridge i Composite su primjenjeni na objektima.
* *Pogled* programski paket sadrži klase koje definišu način predstavljanja podataka iz modela. On je zadužen za grafičko predstavljanje podataka korisnicima. Dizajn šabloni koji su iskorišteni za razradu pogleda su: Composite, Decorator na komponenetama te Command dizajn šablon.
* U slučaju grafičkog editora MiT kontroler je izostavljen zbog prirode samog softvera.
* Observer omogućava definisanje načina pretplate za obavještavanje više objekata o nastalim događajima na objektu koji se posmatra.

# Pogled na procese

Pogled prema procesima opisuje zadatke uključene u izvršenje sistema, njihove interakcije i konfiguracije. Takođe opisuje alokaciju objekata i klasa u zadacima. Obavljanje bilo kog procesa u okviru sistema zasniva se na saradnji klasa.

# Pogled na implementaciju

Implementaciona arhitektura softvera odgovara dizajnu pri čemu se MVC dizajn šablon razlaže u odgovarajuće pakete koji odgovaraju pojedinim funkcijama i koje sadrže odgovarajuće klase potrebne za realizaciju tih funkcionalnosti.

## Slojevi

* Sloj podataka predstavlja sloj koji sadrži informacije o podacima i osnovnim dijelovima softvera koji su u operativnoj memoriji računara. Njegova implementacija se vrši pomoću klasa iz programskog paketa model.
* Prezentacioni sloj predstavlja sloj koji služi za prezentaciju podataka prema korisniku, tj. njihovu grafičku reprezentaciju. Implementira se programskim paketom view. Kako on prikazuje podatke iz sloja podataka mora imati neposrednu svijest o njemu
* U paketu observer je smješten istoimeni obrazac ponašanja. On omogućava definisanje načina pretplate za obavještavanje više objekata o nastalim događajima na objektu koji se posmatra. Prate se promjene na dijagramu i aplikacionom modelu o kojima se obavještava prezentacioni sloj.
* U paketu main je smještena main klasa.

# Veličina i performanse

Izabrana softverska arhitektura podržava ključne zahtjeve po pitanju veličine i performansi, kao što je navedeno u dokumentu Specifikacija Softverskih zahtjeva. Veličina softvera je relativno mala, sa brojem klasa 80-100 koji rezultuju dobrim karakteristikama u pogledu memorijskog iskorišćenja zbog primjenjenih dizajn šablona.

# Kvalitet

Arhitektura softvera podržava zahtjeve kvaliteta, kao što je navedeno u dokumentu Specifikacija softverskih zahtjeva.