

Dr Milunka Damnjanović, red.prof,
**OBJEKTNO ORIJENTISANE TEHNIKE
PROJEKTOVANJA SISTEMA**



7 Interakcija objekata

Dr Milunka Damnjanović, red.prof,
**OBJEKTNO ORIJENTISANE TEHNIKE
PROJEKTOVANJA SISTEMA**

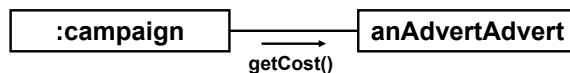


7 Interakcija objekata

Slanje poruka:

Kad jedan objekat šalje poruku drugom objektu, on poziva *operaciju* u pozvanom objektu.

Primer:



Kompleksniji slučaj je kada objekat koji je primio poruku mora da pošalje poruke inicirajući saradnju sa drugim objektima.



Odgovornost klasa #1:

Cilj OO analize i projektovanja je da se funkcionalnost sistema distribuirava ravnomerno po njegovim klasama.

To ne znači da sve klase imaju isti nivo odgovornosti već da svaka klasa treba da ima odgovarajući nivo odgovornosti.

Kada su odgovornosti ravnomerno distribuirane, klase neće postati prekomerno kompleksne i biće olakšan razvoj, testiranje i održavanje.

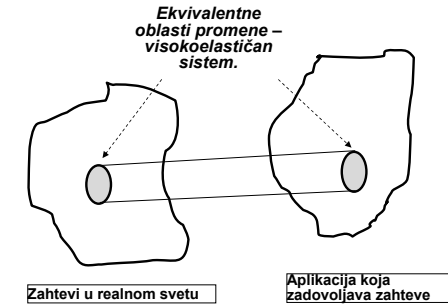
Odgovornost klasa #2:

Klasa koja je relativno mala i samosadržana ima više mogućnosti da bude *ponovo upotrebljena*, nego klasa koja je velika, kompleksna i ima odgovornost ili funkcionalnost koja nije jasno fokusirana.

Pogodna distribucija odgovornosti među klasama rezultuje u sistemu koji je fleksibilniji na promene zahteva.

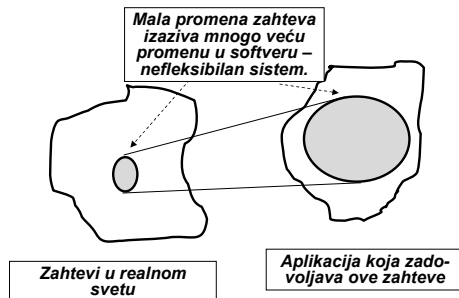
5

Fleksibilnost projekta #1:



6

Fleksibilnost projekta #2:



7

Zadatak modela interakcije:

Osnovni zadatak modela interakcije objekata je definisati najpogodniji način komunikacije porukama između objekata da bi se podržao svaki zahtev posebno.

Korisnički zahtev – dijalog između aktera i sistema koji rezultuje objektima koji izvršavaju zadatke tako da sistem može da odgovori na način koji zahteva akter.

8

Uloga dijagrama interakcije #1:

Dijagrami interakcije sadrže objekte koji predstavljaju korisnički interfejs (*granični objekti*) i koji upravljaju komunikacijom objekata (*kontrolni objekti*).

Ako takvi objekti nisu definisani, u većini slučajeva to znači da je potrebno definisati ih kasnije.

9

Uloga dijagrama interakcije #2:

U toku analize treba identifikovati prirodu dijaloga preko korisnikove potrebe za informacijom i njegovog pristupa funkcionalnosti sistema.

Granični objekti su deo analize i deo projektovanja, pri čemu detaljna specifikacija graničnog objekta koja upravlja dijalogom predstavlja projektnu aktivnost.

10

Saradnja (kolaboracija):

UML razmatra interakciju objekata u kontekstu saradnje koju definiše kao:

Struktura uzoraka koji igraju ulogu u ponašanju i njihovim relacijama (odnosima) naziva se saradnjom ili kolaboracijom.

Pod ponašanjem se podrazumeva operacija, ili korisnički slučaj, ili bilo koji klasifikator ponašanja u UML-u.

11

Sekvencijalni dijagrami interakcije:

Sekvencijalni dijagram interakcije (ili samo *sekvencijalni dijagram, SD*) je jedan od dva tipa dijagrama interakcije u UML-u (drugi je dijagram saradnje - kolaboracije koji je već razmatran).

12

Interakcija:

Interakcija se definiše u kontekstu *saradnje* (kolaboracije). Ona specificira oblike komunikacije između uloga u saradnji. Preciznije, ona sadrži skup parcijalno uređenih *poruka* gde svaka poruka specificira jednu komunikaciju (npr. koji signal treba poslati ili koju operaciju treba pozvati).

13

Sekvencijalni dijagram:

Sekvencijalni dijagram prikazuje interakciju između objekata uređenih u vremensku sekvencu.

Najopštija primena sekvencijalnih dijagrama je predstavljanje detaljne interakcije objekata koje se pojavljuju u jednom korisničkom slučaju ili jednoj operaciji.

14

SD u analizi i projektovanju:

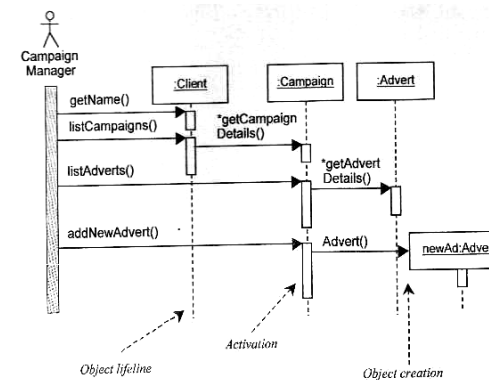
SD crtani u toku analize razlikuju se od SD-a crtanih u toku projektovanja u dva pogleda:

Analički sekvencijalni dijagrami normalno ne sadrže projektne objekte niti specificiraju detalje signature poruke.

15

Osnovni koncepti i notacija:

Sekvencijalni dijagram SD1 za korisnički slučaj dodavanja nove reklame u kampanji (Add New Advert):



16

Osnovni koncepti i notacija:

- Vertikale predstavljaju tok vremena,
- Horizontale predstavljaju objekte u interakciji.
- Svaki objekat predstavljen je vertikalnom isprekidanom linijom – *lifeline*, sa simbolom objekta na vrhu
- Poruka je predstavljena punom horizontalnom linijom i ima ime.

17

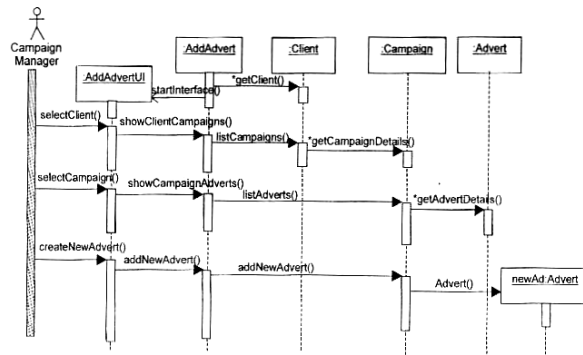
Stimulus, signal, događaj:

- Stimulus (u UML-u) – interakcija dva objekta koji prenose informaciju očekujući akciju. (Opisani su pošiljalac, primalac i akcija.)
- Signal (u UML-u) – asinhrona komunikacija koja može imati parametre.
- Događaj (u UML-u) – pojava nečeg značajnog.

18

SD1 sa graničnim i kontrolnim klasama:

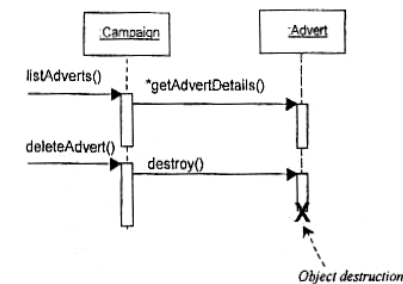
Sekvencijalni dijagram SD2 za korisnički slučaj dodavanja nove reklame u kampanji (Add New Advert) sa graničnim i kontrolnim klasama:



9

Destrukcija objekta:

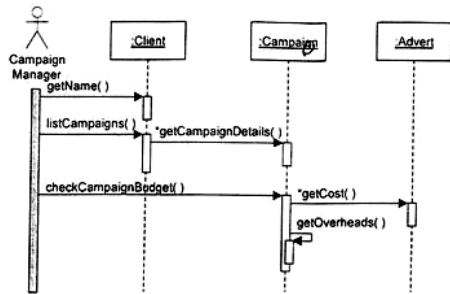
Objekat se može formirati i ukloniti na bilo kom stepenu u toku interakcije.



20

Refleksivna poruka:

Refleksivna poruka je poruka koju objekat šalje sam sebi. Primer:



21

Fokus kontrole:

Fokus kontrole prikazuje trenutke u toku aktivacije (izvršenja operacije zahtevane porukom) kada se dešava obrada unutar objekta.

Fokus kontrole može se prikazati senčenjem dela pravougaonika aktivacije koji odgovara aktivnoj obradi jedne operacije.

22

Povratak (return):

Povratak (return) je povratak kontrole objektu od koga je potekla poruka koja je započela aktivaciju.

(To nije nova poruka već završetak poziva operacije.)

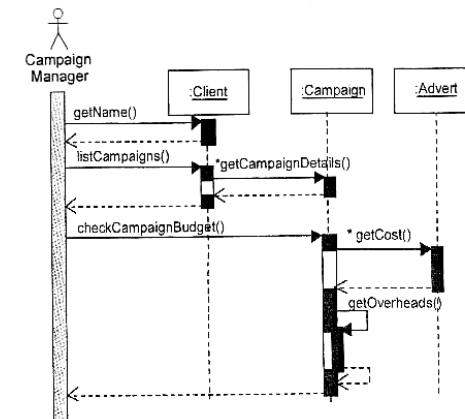
Povratak se prikazuje isprekidanom strelicom koja je opciona jer se povratak kontrole podrazumeva.

Vraćena veličina je veličina koju operacija vraća objektu koji je pozvao.

23

Primer SD sa fokusom kontrole:

SD za proveru budžeta kampanje (Check campaign budget):



24

Sinhrona poruka:

Sinhrona poruka ili *proceduralni poziv* izaziva da pozvana operacija suspenduje izvršavanje dok joj se fokus kontrole ne vrati.

Proceduralni pozivi su pogodni za interakciju jer svaka operacija koja zove drugu, čini to da bi dobila podatke i ne može da nastavi dok ne dobije podatke.

Predstavlja se strelicom sa punom glavom.

25

Asinhrona poruka:

Asinhrona poruka ne izaziva prekid izvršenja dok se čeka na povratak.

Kada se pošalje asinhrona poruka, operacije u oba objekta mogu da vrše obradu istovremeno.

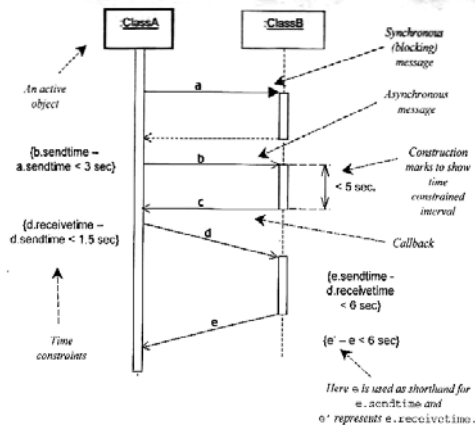
Asinhrona poruka često se koriste kod real-time sistema gde operacije u različitim objektima moraju da se izvršavaju konkurentno ili zbog efikasnosti ili zato što simuliraju aktivnosti iz stvarnog sveta koje se odvijaju konkurentno.

Neophodno je da operacija koja je pozvana asinhrono obavesti objekat koji je pozvao da je završila rad. To se radi slanjem poruke (poznate kao callback) objektu koji je pozvao.

Predstavlja se strelicom sa otvorenom glavom.

26

SD sa različitim tipovima poruka:



27

Vremenska ograničenja:

SD može imati oznake koje dokumentuju vremenska ograničenja.

Vremenska ograničenja mogu biti pridružena npr. opisu akcije ili izvršenju operacije.

Vremenski izrazi mogu biti pridruženi imenu poruke.

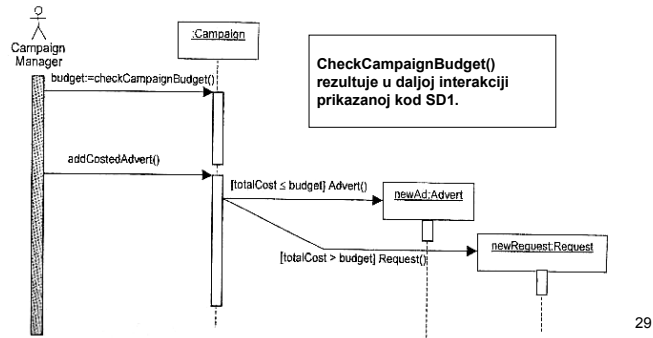
Vreme potrebno za slanje poruke zanemaruje se u odnosu na vreme izvršenja jer je mnogo kraće.

Ako je vreme slanja poruke (npr. kod distribuiranih sistema), strelica koja predstavlja poruku crta se koso nadole

28

Grananje:

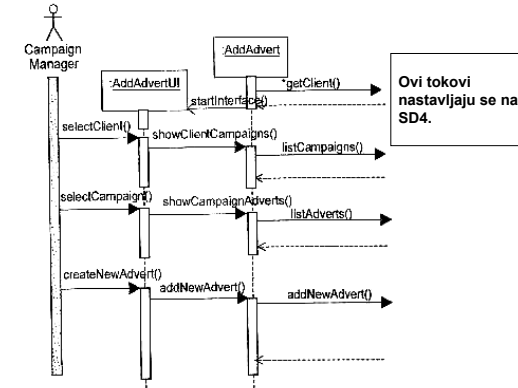
Neke interakcije imaju dva ili više alternativnih puteva. Svaki predstavlja granu u mogućem nizu događaja kod korisničkog slučaja.



29

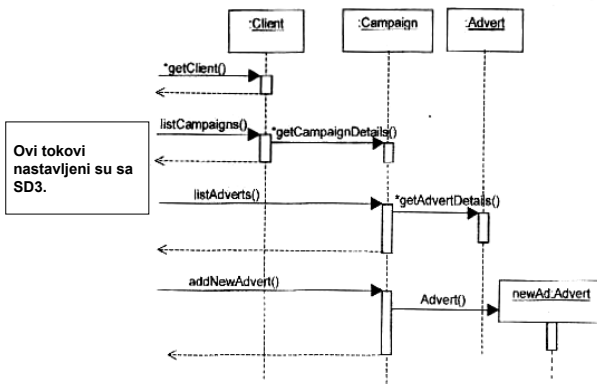
Upravljanje SD-om:

Moguće je od dva ili više SD-a formirati jedan, a moguće je i jedan SD razbiti na dva ili više manjih. Primer SD3:



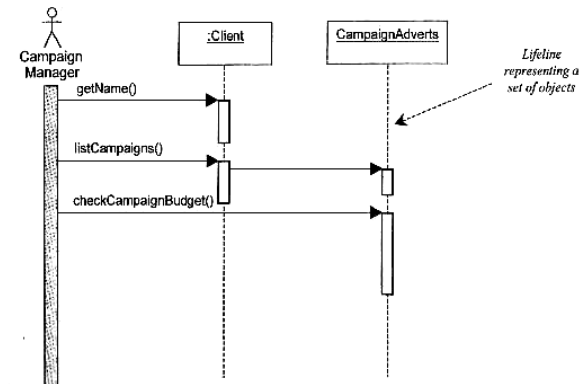
30

Upravljanje SD-om:



31

Alternativni SD sa grupisanjem objekata, SD5:



32

Modelovanje real-time sistema i konkurentnosti #1:

Real-time sistemi imaju glavnu osobinu da moraju da odgovore na spoljašnje događaje u strogo ograničenom vremenu. Zato oni često pokazuju konkurentno ponašanje u formi puteva simultanog izvršavanja ili *kontrolnih niti (threads of control)*.

Aplikacija koja ima konkurentno izvršavanje uvek sadrži neke objekte koji koordiniraju i iniciraju kontrolne niti. Oni su poznati kao *aktivni objekti* i uzorci aktivnih klasa. Mnogo drugih objekata radi unutar kontrolne niti i oni su poznati kao *pasivni objekti* i pripadaju pasivnim klasama.

Aktivni objekti ili klase predstavljaju se debelom granicom na dijagramu interakcije i često su kompoziti sa ugrađenim delovima.

33

Modelovanje real-time sistema i konkurentnosti #2:

SD konkurentnog sistema prikazuje koje kontrolne niti su aktivne u svakom trenutku. Kontrola se predstavlja sinhronim i asinhronim porukama.

U ovoj primeni, puna (sinhrona) glava strelice predstavlja pridruživanje kontrole niti, a otvorena ili priljubljena (asinhron) glava strelice predstavlja poruku poslata bez pridruživanja kontrole.

34

Dijagrami saradnje (kolaboracije):

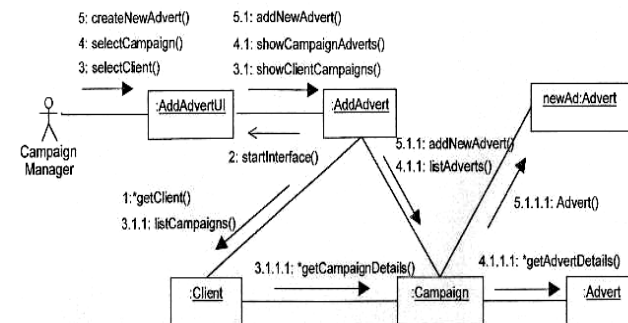
Dijagrami saradnje (DS) imaju mnogo sličnosti sa sekvencijalnim dijagramima (SD). Zato se DS mogu koristiti za automatsko generisanje SD i obrnuto.

Najznačajnija razlika između DS i SD je što DS eksplicitno prikazuje veze između objekata koji učestvuju u saradnji. Nasuprot SD-u, kod DS-a nema eksplicitne vremenske dimenzije.

35

Primer DS1:

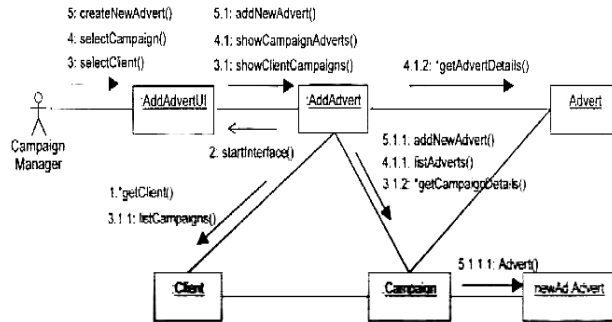
Dijagram saradnje DS1 za korisnički slučaj dodavanja nove reklame u kampanji (Add New Advert):



36

Primer DS2:

Modifikovani DS1:



37

Oznake poruka na DS-u:

Oznake poruka – skup simbola (kao kod sd-a).

Prethodnik (predecessor) je lista niza brojeva poruka koje moraju da se pojave pre nego tekuća poruka dobije dozvolu. Ovo omogućava detaljno predstavljanje puteva grananja.

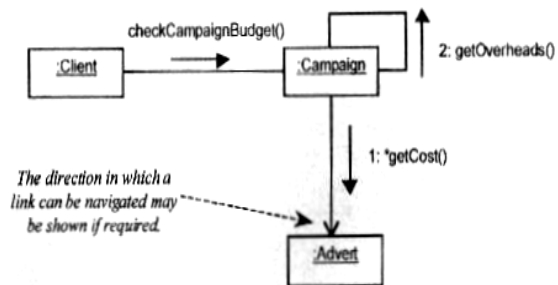
Vodeći uslovi (guard conditions) koriste se za predstavljanje sinhronizacije različitih kontrolnih niti.

Sekvencijalni izraz je lista celih brojeva odvojenih tačkom ‘.’ iza koga opciono sledi *ime* (jedno slovo), iza koga opciono sledi termin *recurrence* i završava se dvotačkom ‘:’.

38

Primer DS3:

Dijagram saradnje za korisnički slučaj za proveru budžeta kampanje (Check campaign budget):



39

Kratak sadržaj:

- Interakcija objekata je bitno svojstvo OO pristupa razvoju sistema.
- UML koristi dve ekvivalentne tehnike modelovanja interakcije – sekvencijalne dijagrame (SD) i dijagrame saradnje (DS).
- DS se najčešće koriste pri analizi korisničkih slučajeva.
- SD se može koristiti za efektno predstavljanje detaljnih specifikacija interakcija u projektu.
- Kod kompleksnih interakcija mogu se koristiti više povezanih (linkovanih) dijagrama.

40