Principi softverskog inženjerstva

Vežbe - X nedelja
Modelovanje Web aplikacije na UML-u

Dražen Drašković, asistent
Elektrotehnički fakultet
Univerziteta u Beogradu

UML

- UML (engl. Unified Modeling Language) - Objedinjeni jezik za modelovanje
- Grafički jezik za vizualizovanje, specifikovanje, konstruisanje i dokumentovanje u nekom softverskom sistemu
- Standard za šematsko prikazivanje sistema, koji obuhvata i koncepcijske aspekte (poslovni procesi i funkcije sistema) i konkretne aspekte (klase u nekom prog.jeziku, šeme baza podataka, softverske komponente koje se mogu lako ponovo koristiti)

Dijagram

- Dijagram je grafička reprezentacija skupa povezanih elemenata. Dijagram se najčešće pojavljuje u obliku grafa čvorova (stvari) povezanih granama (relacijama).
- Dijagrami se crtaju da bi se sistem vizualizovao iz različitih perspektiva.

Vrste dijagrama

- Postoje:
  - statički dijagrami
  - dinamički dijagrami

Statički dijagrami

- Dijagram klasa (*class diagram*) prikazuje logičku strukturu apstrakcija: skup klasa, interfejsa, kolaboracija i njihovih relacija.
- Dijagram objekata (*object diagram*) prikazuje logičku strukturu instanci: skup objekata (instanci klasa) i njihovih veza.
- Dijagram komponenata (*component diagram*) prikazuje fizičku organizaciju i zavisnosti između skupa komponenata.
- Dijagram raspoređivanja (*deployment diagram*) prikazuje konfiguraciju čvorova obrade i komponenata koje žive na njima.
- Dijagram paketa (*package diagram*) prikazuje statičku strukturu grupisanja elemenata modela u pakete.
- Dijagram složene strukture (*composite structure diagram*) prikazuje hijerarhijsko razlaganje složene klase (objekta) na delove.

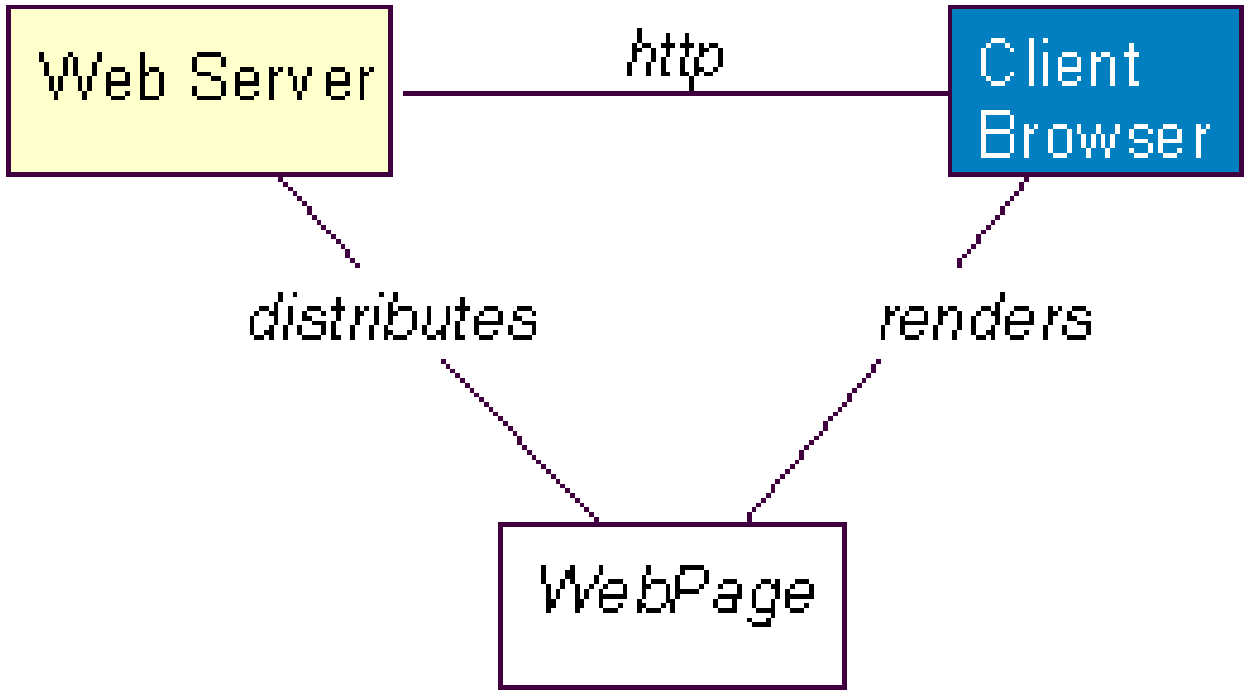
Dinamički dijagrami

- Dijagram slučajeva korišćenja (*use case diagram*) prikazuje skup slučajeva korišćenja, aktera (specijalne vrste klasa) i njihovih relacija
- Dijagram interakcije (*interaction diagram*) prikazuje jednu interakciju koju čine skup objekata i njihovih veza sa porukama koje razmenjuju
  - Dijagram sekvence (*sequence diagram*) je dijagram interakcije koji naglašava vremenski redosled poruka
  - Dijagram komunikacije (*communication diagram*) je dijagram interakcije koji naglašava strukturnu organizaciju objekata koji šalju i primaju poruke;
  - Dijagram pregleda interakcije (*interaction overview diagram*) [UML 2.0] je dijagram interakcije koji definiše interakcije kroz vrstu dijagrama aktivnosti (kombinacija d. aktivnosti i d. sekvence)
  - Vremenski dijagram (*timing diagram*) [UML 2.0] je dijagram interakcije koji prikazuje promenu stanja objekata u vremenu
- Dijagram aktivnosti (*activity diagram*) prikazuje tok od jedne do druge aktivnosti u sistemu (nije specijalna vrsta dijagrama stanja u UML2.0)
- Dijagram stanja (*statechart diagram*) prikazuje konačni automat koji obuhvata stanja, tranzicije, događaje i aktivnosti

Arhitektura Web aplikacije
Web sajt

- Web sajt sadrži tri glavne komponente:
  - web server,
  - mrežnu konekciju,
  - jedan ili više klijentskih browser-a.
- Web server distribuira (web) strane formatiranih informacija klijentima koji ih zahtevaju.
- Zahtev se postavlja preko mrežne konekcije i upotrebljava HTTP protokol.

Arhitektura Web aplikacije



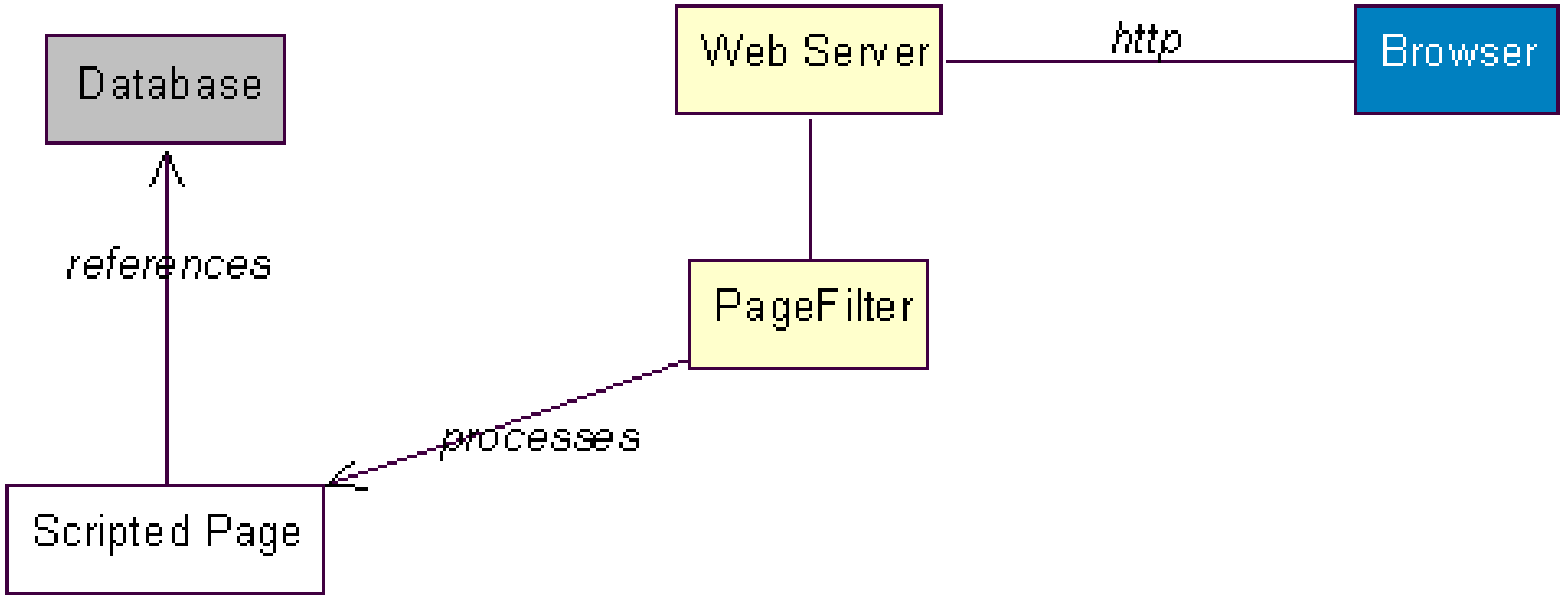
Osnovna arhitektura Web aplikacije

- Informacije koje pruža Web sajt tipično su zapamćene u formatiranom obliku, u fajlovima
- Klijent zahteva fajl po imenu i kada je neophodno pruža i informaciju o njegovoj celokupnoj putanji (adresi). Ovi fajlovi se nazivaju stranicama i reprezentuju sadržaj Web sajta.

Osnovna arhitektura Web aplikacije

- U nekim situacijama sadržaj stranice nije statički određen (memorisan u fajlu), već se sklapa dinamički od informacija iz baze podataka (ili drugih repozitorijuma informacija) i formatira na osnovu niza instrukcija (skripta) koji se čuva u fajlu.
- Web server upotrebljava filter (intepreter) stranica da interpretira i izvrši skriptove.
- Web sajtovi koji primenjuju ovu strategiju nazivaju se dinamički sajtovi.

Arhitektura dinamičkog Web sajta



Skriptovanje na serverskoj strani

- Krajnji rezultat serverskog procesiranja je:
  - ažuriranje stanja na serveru
    (baza podataka, poslovni objekti),
  - priprema stranice formatirane u HTML-u
    (korisničkog interfejsa) za klijentski browser koji je postavio zahtev.

Skriptovanje na klijentskoj strani

- Klijentski browser takođe može da izvršava programski skript na stranici.
- Primer: JavaScript
- Kada browser izvršava takav skript, on nema neposredan pristup serverskim resursima (baza podataka).
- Skriptovi koji se izvršavaju na klijentu tipično dopunjavaju izgled i ponašanje korisničkog interfejsa, a ne implementiraju poslovnu logiku aplikacije.

Forme (1)

- Najviše korišćeni mehanizam za sakupljanje ulaznih podataka od korisnika je putem HTML formi.
- HTML forma je kolekcija ulaznih polja koji se prikazuju (render-uju) kao web stranica. Osnovni ulazni elementi su:
  - tekstualno polje (text box),
  - tekstualna oblast (text area),
  - polje za čekiranje (checkbox),
  - grupa radio dugmadi (radio button group),
  - selekciona lista (selection list).

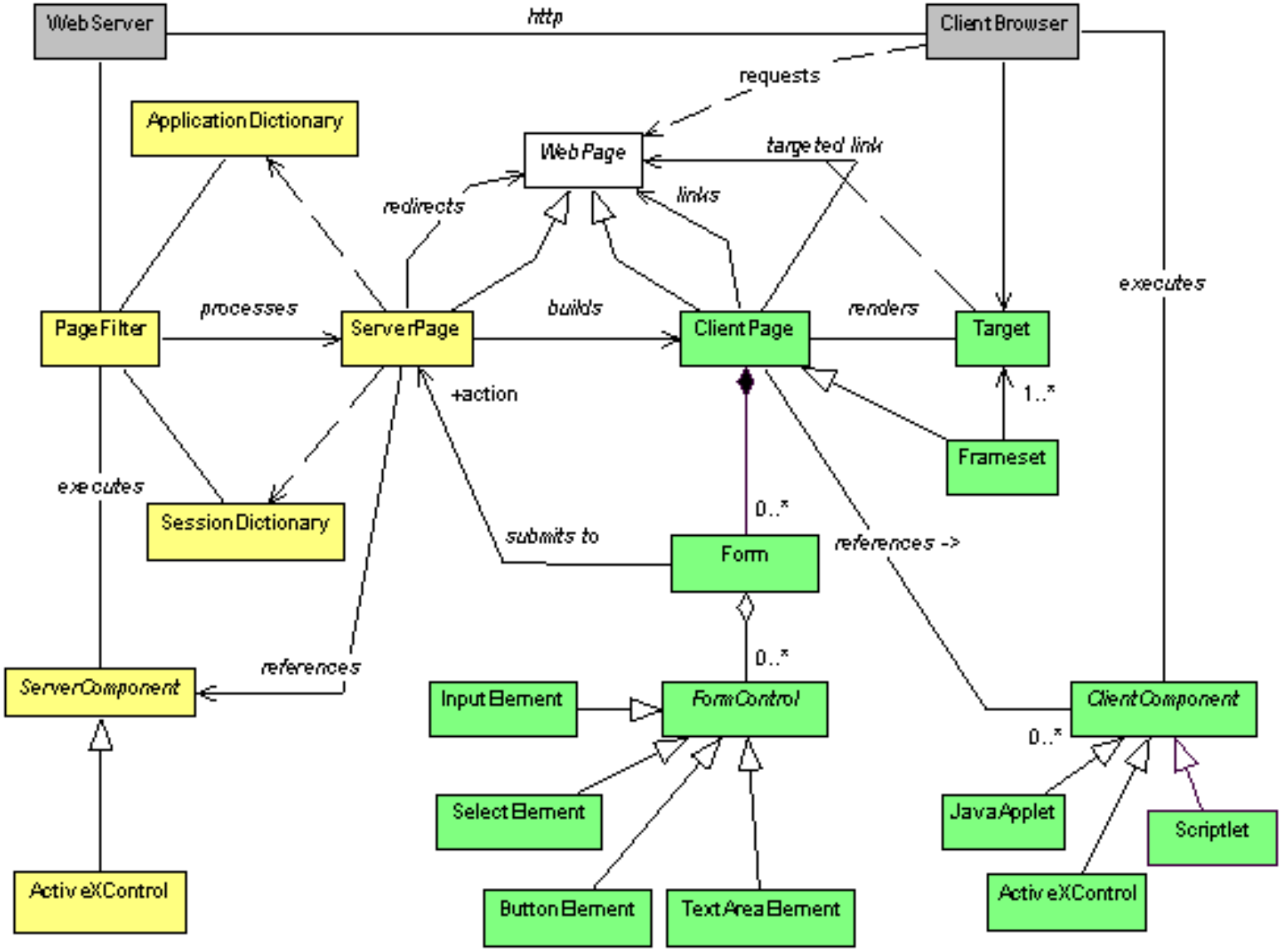
Forme (2)

- Svi ulazni elementi na formi identifikovani su imenom (name) ili identifikatorom (id).
  `<input type="text" name="username">`
- Svaka forma se asocira sa akcionom stranicom (tako što se navede URL te stranice).
  `<form name="forma" action="glavna.php" method="POST">`
- Akciona stranica služi da primi i procesira informacije sadržane u popunjenoj formi. Skoro uvek reč je o stranici koja sadrži skript za procesiranje.

Forme (3)

- Kada popuni formu, korisnik podnosi (submits) formu na server zahtevajući akcionu stranicu sa servera.
- Web server pronalazi tu stranicu i interpretira (izvršava) sadržani programski skript.
- Programski skript može da čita informacije koje su podnete sa forme.

Model generalizovane arhitekture



Modelovanje Web aplikacije

- Kako modelovati stranicu?

  Koristeći UML možemo da izrazimo stranicu kao objekat. To onda dovodi do pitanja, šta su svojstva tog objekta?

  Da li je prikladno da izrazi elemente rasporeda (fontovi, tabele, tekst, itd)? Ukoliko postoje skriptovi na stranici, da li mogu da se identifikuju kao metode objekta?

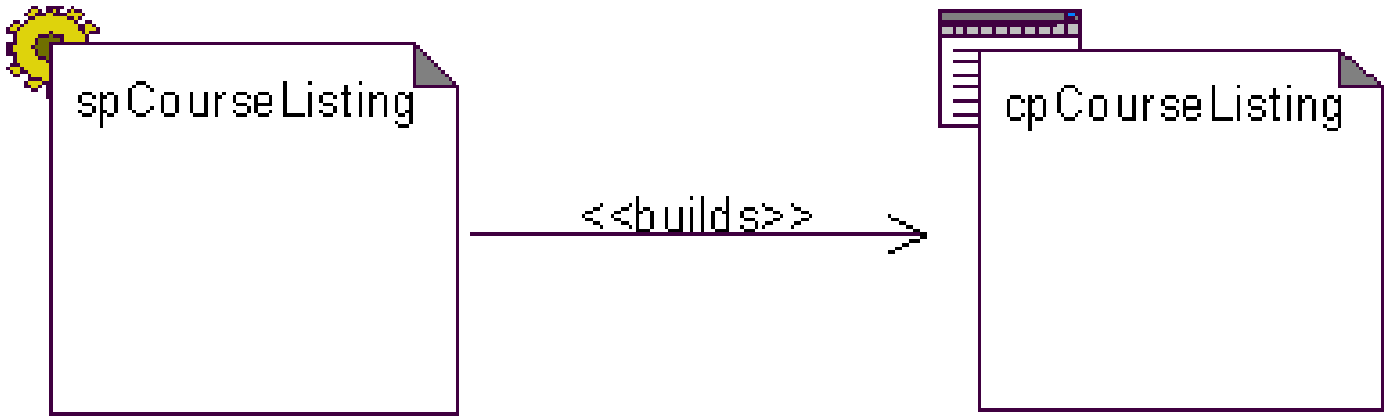
Modelovanje Web stranice (1)

- U opštem slučaju, proizvoljna web stranica u web aplikaciji koja ima neku funkcionalnost i na serveru i na klijentu u UML modelu se predstavlja sa dve različite klase, iako je implementacija stranice najčešće u jednom fajlu.
- U ovoj situaciji metodi i promenljive serverskog programskog skripta sadrže se u modelu u **klasi** koja ima **stereotip «server page»**.

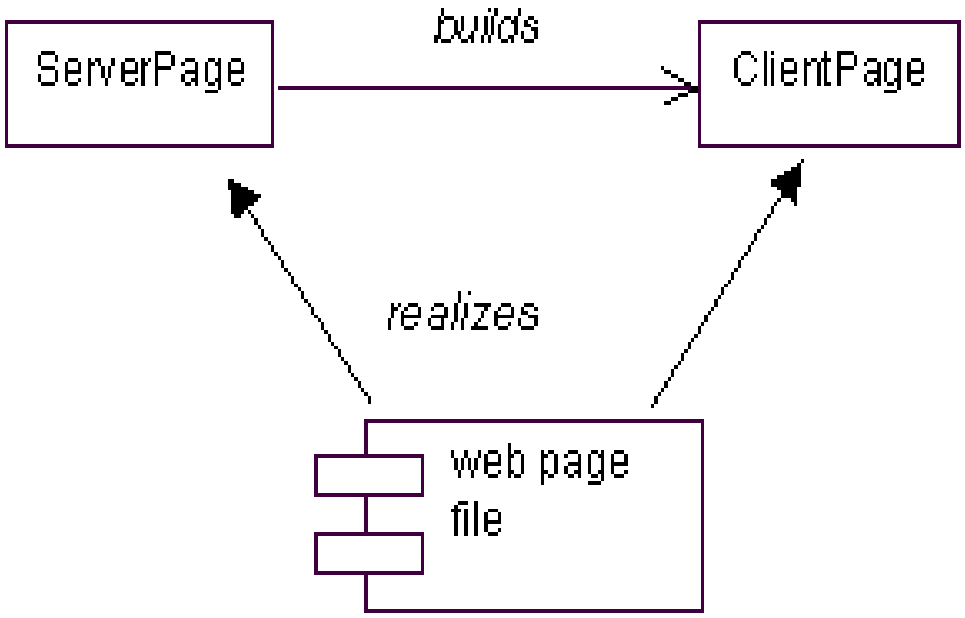
Modelovanje Web stranice (2)

- Klijentske stranice se na sličan način u UML modelu reprezentuju **klasom** sa **stereotipom «client page»**. Atributi klase su promenljive i funkcije koje izvršava klijentski browser.
- Osnovna relacija između klijentske i serverske klase određene web stranice je ***gradi*** (***builds***). Serverska stranica ***gradi*** klijentsku stranicu. Da bi se ova relacija predstavila, od serverske klase iscrtava se jednosmerna asocijacija sa stereotipom **«builds»** ka klijentskoj klasi.

Serverska stranica gradi klijentsku

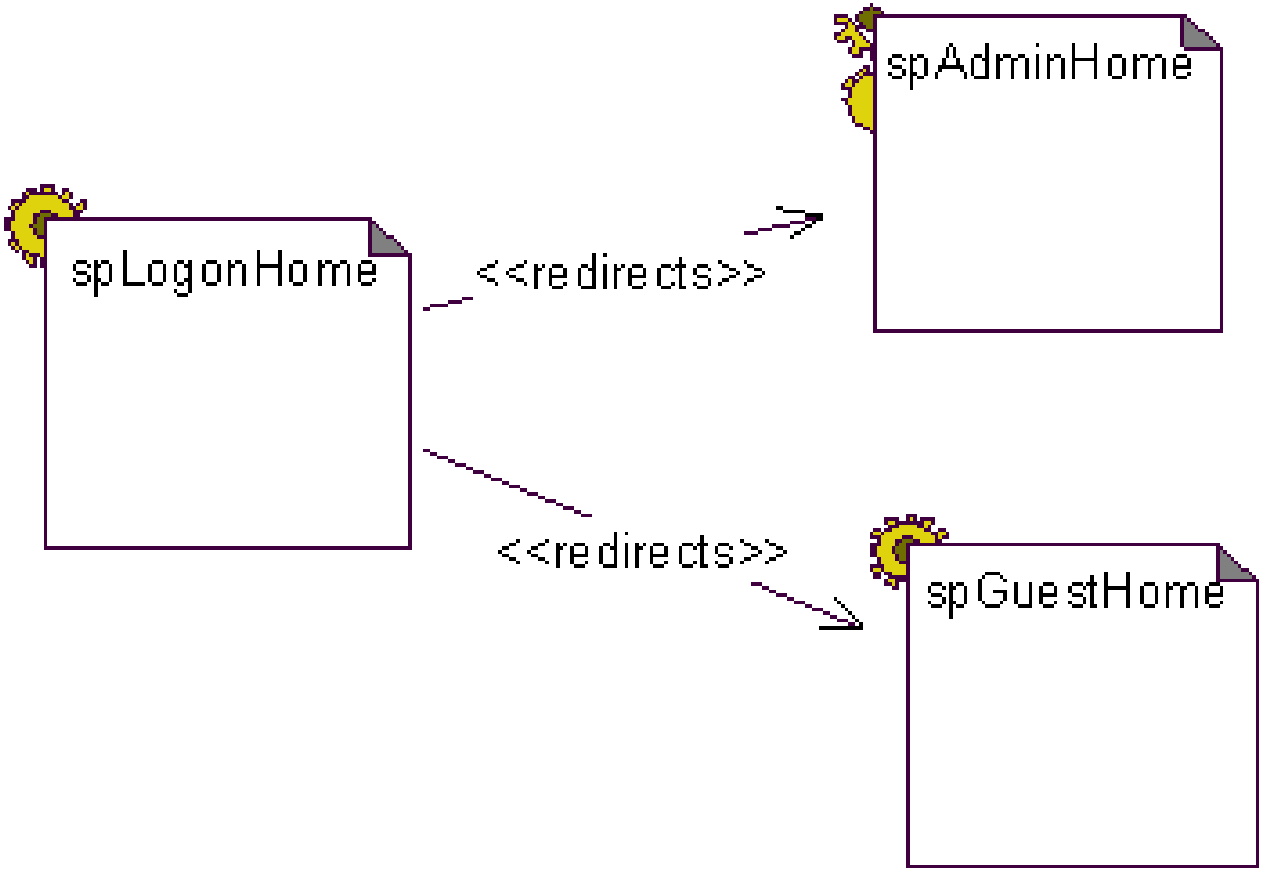


Web stranica = server + klijent str.



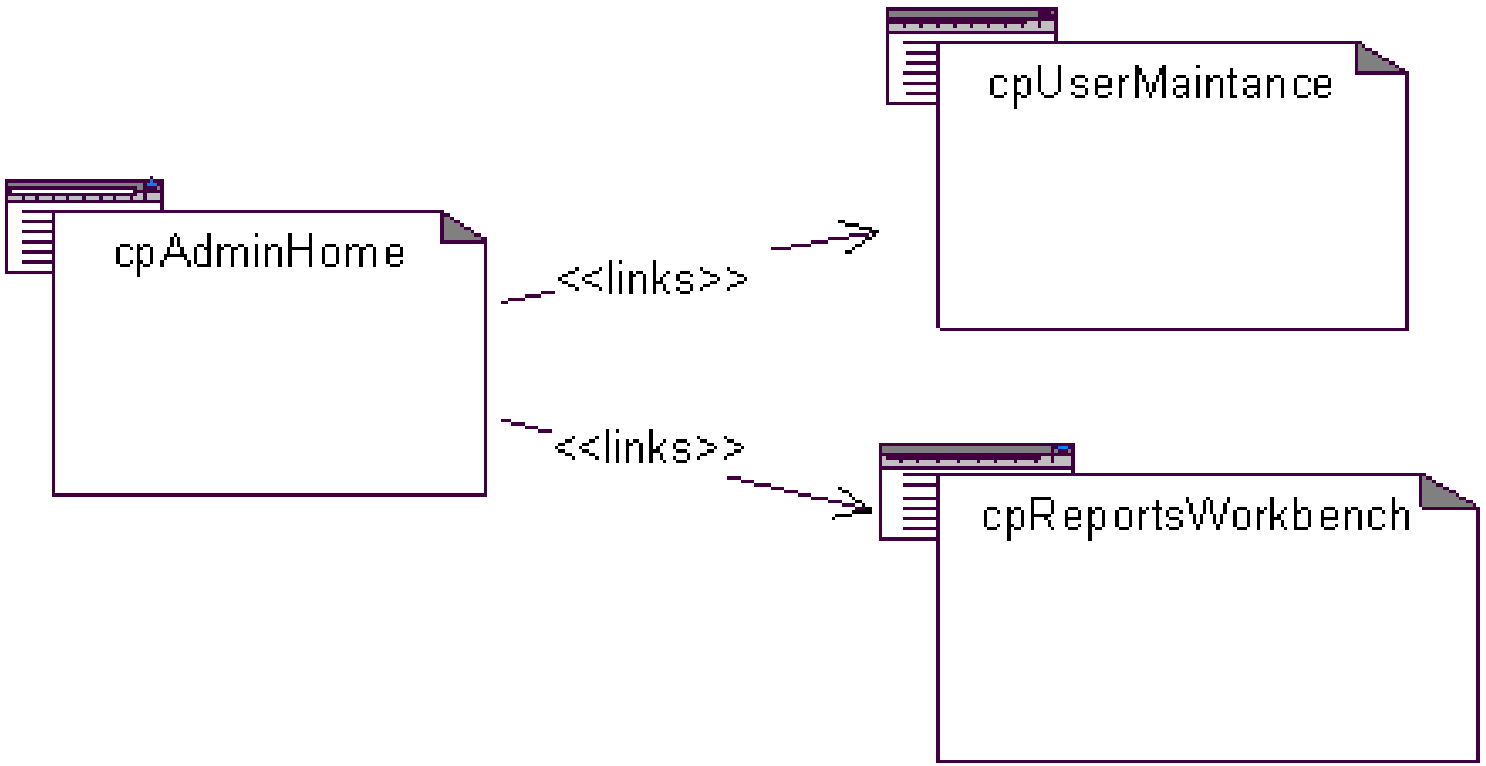
Serverska stranica delegira posao drugoj stranici

- Neke tehnologije za razvoj Web aplikacija omogućavaju preusmeravanje (redirection) klijentskog zahteva (i podataka) od jedne serverske stranice drugoj. Ova relacija u modelu se predstavlja **asocijacijom** sa stereotipom **«redirects»**.
- Ovo se, na primer, može koristiti u varijanti strukture Web aplikacije gde jedna serverska stranica prihvata sve korisničke zahteve a zatim ih redirektuje za dalje procesiranje.

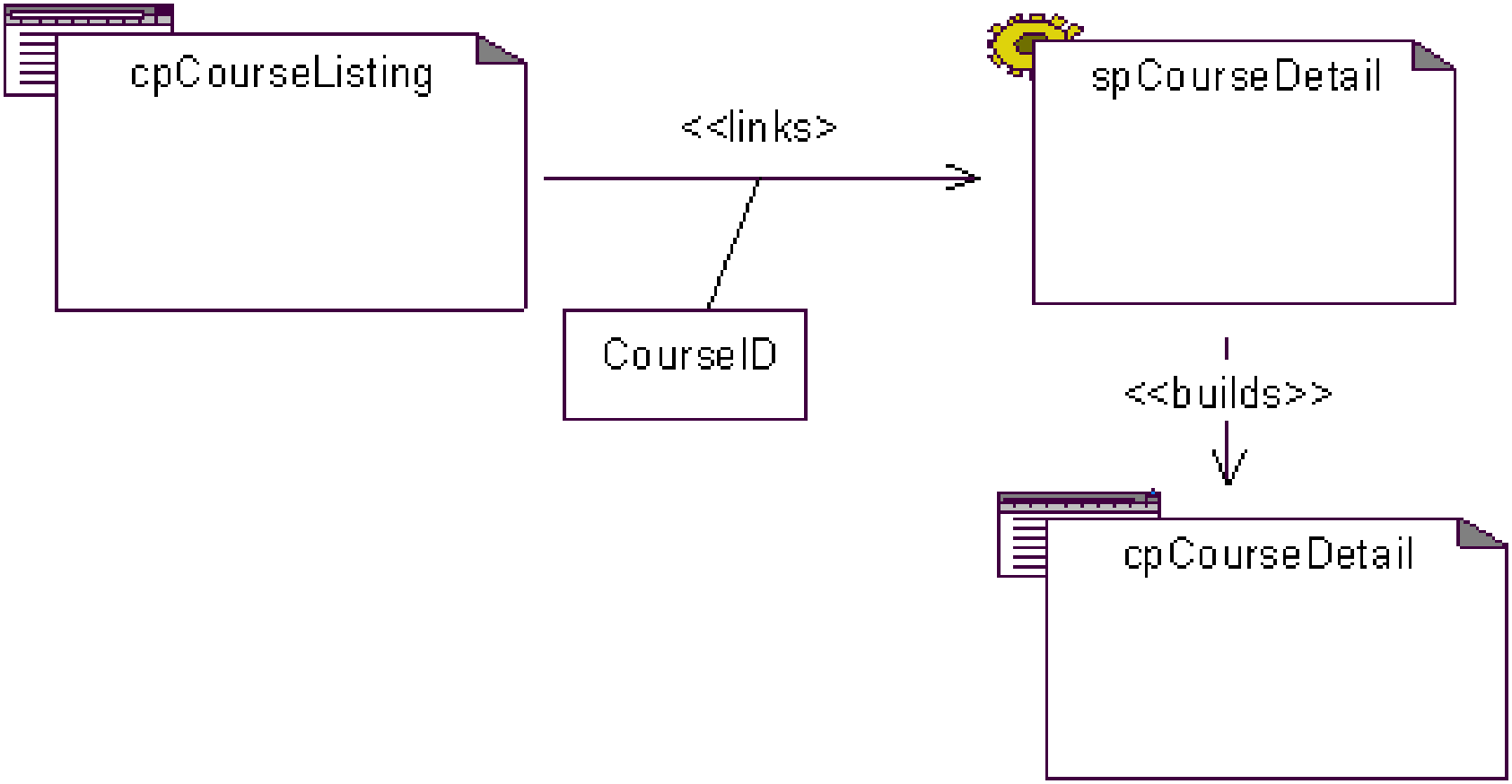

spAdminHome
spLogonHome
<<redirects>>
<<redirects>>
spGuestHome

Klijentske stranice međusobno povezane hiperlinkovima

- Klijentske stranice često sadrže **hiperlinkove** (ili anchors) ka drugim web stranicama. Te druge stranice mogu biti serverske ili klijentske (njih naravno zahteva browser kada se klikne na hiperlink).
- Stereotype: **«links»** se primenjuje na asocijacije između klijentskih stranica i drugih (serverskih ili klijentskih) stranica. Besmisleno je da izvor relacije **«links»** bude serverska stranica.


cpUserMaintance
cpAdminHome
<<links>>
<<links>>
cpReportsWorkbench

Povezivanje sa prosleđivanjem parametara



Forme (1)

- Ako imamo više formi na jednoj istoj Web stranici, pri čemu svaka forma cilja različitu *akcionu* stranicu. Ovo se može modelovati kreiranjem nove klase da reprezentuje HTML formu (deo tekstualnog sadržaja stranice), sa stereotipom **«form»**.

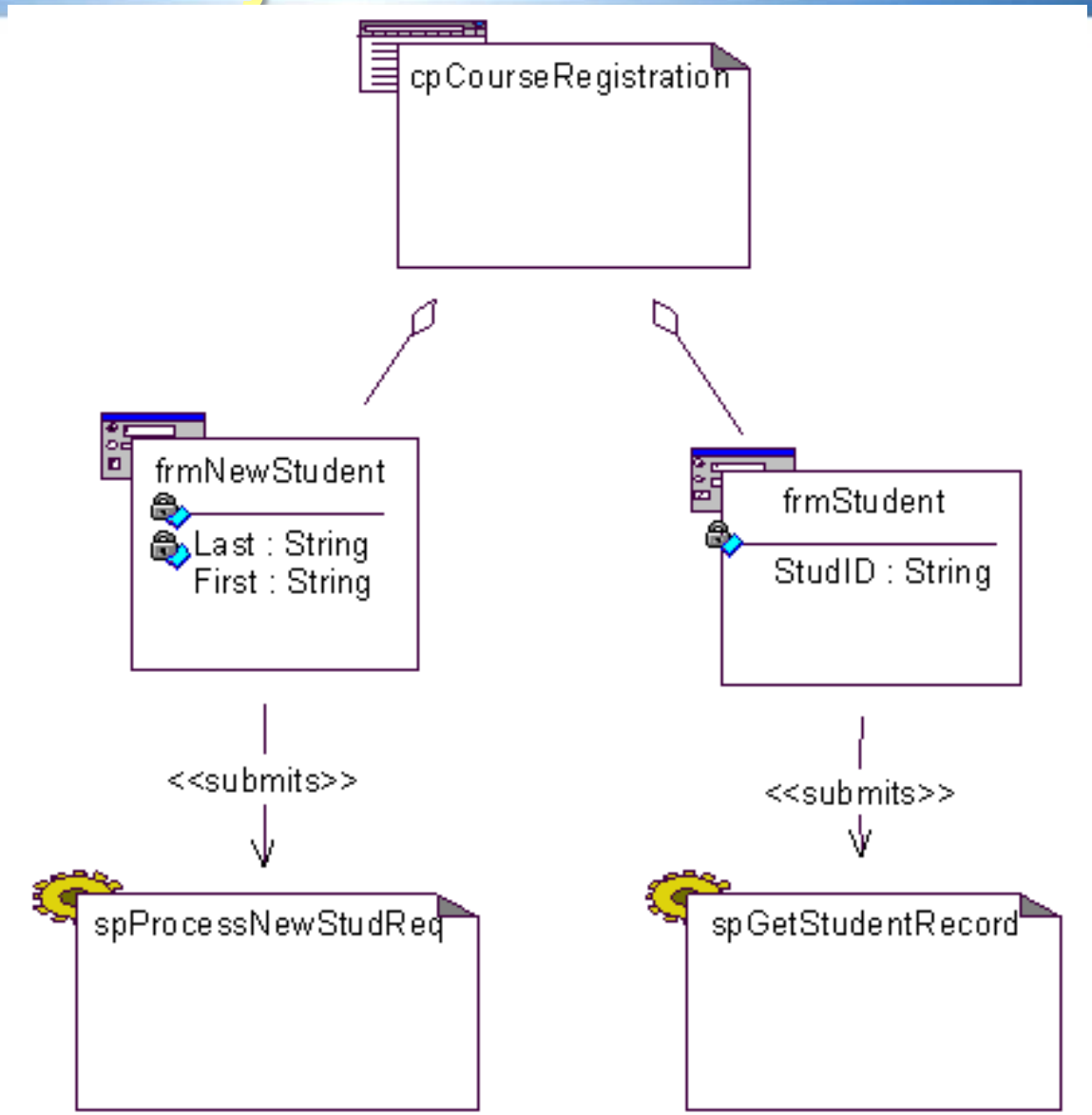
Forme (2)

- Atributi klase koja predstavlja HTML formu su polja te forme. Klase formi ne sadrže metode, jer metoda definiše dinamičko ponašanje u kontekstu jedne forme. Metode se stavljaju u klasu klijentske stranice u okviru koje se forme nalaze. Na ovaj način, metode klijentske stranice imaju pristup atributima svih formi koji se nalaze na istoj stranici. Ispravna relacija između klijentske stranice i forme je relacija sadržanja. Klijentska stranica sadrži formu. Na UML dijagramu se to predstavlja agregacijom.

Forme (3)

- Svaka forma je povezana sa nekom Web stranicom (skoro uvek je to serverska stranica) koja je zadužena za prijem i obradu podataka iz forme. Ta stranica se naziva akcionom stranicom.
- Od klase forme do klase akcione stranice se povlači jednosmerna asocijacija sa stereotipom «submits».

Prosleđivanje podataka sa forme ka serverskoj stranici



Ostali stereotipovi

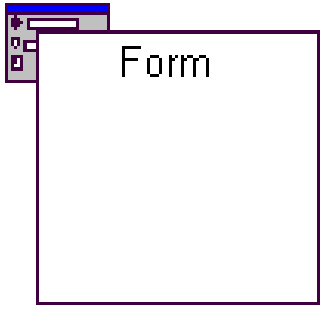
- Kada se koristi skriptovanje na klijentskoj strani, klijentska stranica može da inicira otvaranje prozora za dijalog, da bi se od korisnika pokupili podaci ili korisnik obavestio o nečemu. Ovaj scenario se modeluje jednosmernom asocijacijom od pozivajuće klijentske stranice do neke druge web strane (serverske ili klijentske). Ta asocijacija ima stereotip «dialog», koji govori da je stranica koja je inicirala dijalog privremeno suspendovana sve dok se pozvani prozor za dijalog ne zatvori.

Stereotype Icons

- Serverska stranica

- Klijentska stranica

- Forma



Dijagram interakcije

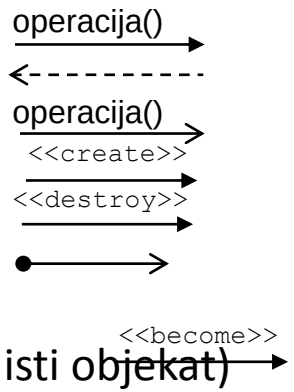
- Interakcija je ponašanje koje obuhvata skup poruka koje se razmenjuju između skupa objekata u nekom kontekstu sa nekom namenom
- Poruka je specifikacija komunikacije između objekata koja prenosi informaciju
  - poruka može biti
    - asinhrona (slanje signala)
    - sinhrona (poziv operacije)
  - od poruke se očekuje da će uslediti aktivnost

Vrste dijagrama interakcije

- Dijagrami interakcije mogu biti:
  - dijagrami sekvence
  - dijagrami komunikacije
  - dijagrami pregleda interakcije (UML 2)
  - vremenski dijagrami (UML 2)
- Dijagrami sekvence naglašavaju vremensko uređenje interakcije
- Dijagrami komunikacije naglašavaju strukturu veza između učesnika u interakciji
  - ove dve vrste dijagrama vizuelizuju na različit način iste informacije
  - semantički su ekvivalentni (izomorfni) i mogu se automatski konvertovati jedan u drugi
- Dijagrami pregleda interakcije kombinuju dijagram aktivnosti sa dijagramima sekvence
  - u okviru toka kontrole, blokovi (čvorovi) se opisuju interakcijama
- Vremenski dijagrami prikazuju promenu stanja jednog objekta (ili uloge) u vremenu
  - promena stanja se dešava kao posledica prijema stimulusa i dešavanja događaja

Slanje i prijem poruke (1)

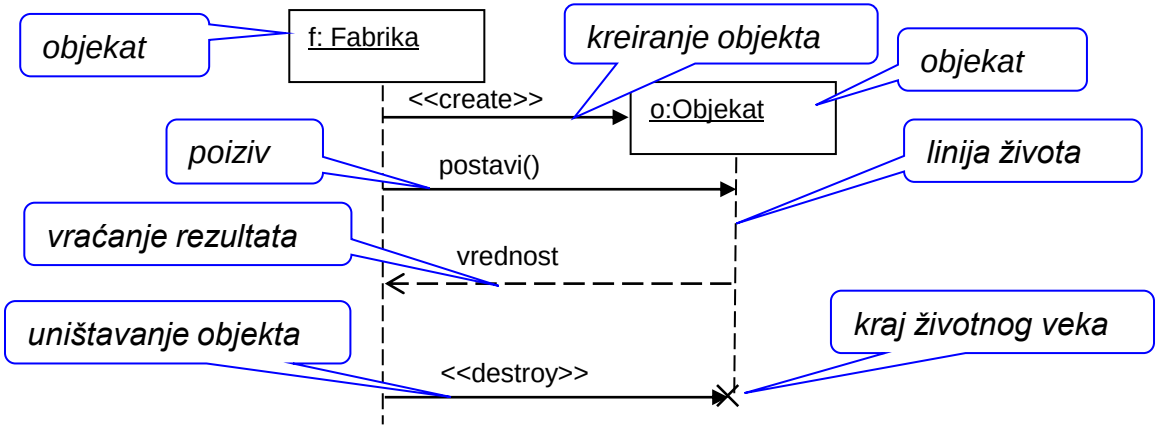
- Prijem jedne poruke se može smatrati instancom jednog događaja
- Kada se pošalje poruka, sledi akcija
  - izvršenje naredbe koja predstavlja apstrakciju metoda
- UML predviđa sledeće vrste poruka:
  - poziv (*call*) – pokreće operaciju objekta primaoca (može biti i poziv sebi)
  - povratak (*return*) – vraća vrednost pozivaocu
  - slanje (*send*) – asinhrono se šalje signal primaocu
  - kreiranje (*create*) – kreira se objekat
  - uništavanje (*destroy*) – uništava se objekat
  - pronađena poruka (*found*) – poznat primalac, slanje nije opisano
  - izgubljena poruka (*lost*) – poznat pošiljalac, prijem se nije dogodio
  - postajanje (*become*) – objekat menja prirodu (na obe strane veze je isti objekat)



- Poruke su horizontalne, jer se podrazumeva atomičnost stimulusa koji predstavlja poruka
  - ako se stimulus ne može smatrati atomičnim – poruka može biti crtana i ukoso naniže

Slanje i prijem poruke (2)

- Kod poruke vrste poziva (*call*) podrazumeva se "sinhronost":
  - pozivalac ne napreduje dok pozvani objekat ne završi obradu poziva
  - cela sekvenca ugneždenih poziva se završava pre nego što se spoljašnji nivo izvršenja nastavi
  - koristi se za proceduralne pozive u jednoj niti, ali i za sinhrone pozive u višeprocesnom okruženju (pozivalac čeka da pozvani objekat obradi poziv)
- Primer dijagrama sekvence i raznih vrsta poruka

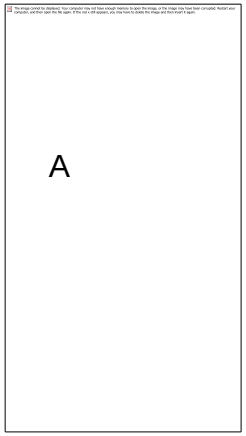


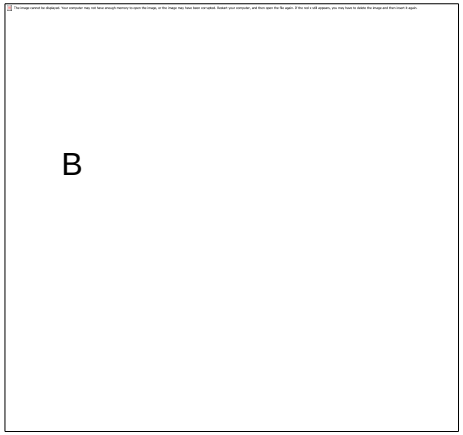
Sintaksa poruke (pojednostavljena)

- Na poruci se osim imena mogu prikazati i
  - njeni argumenti
  - vraćena vrednost
  - pridruživanje vraćene vrednosti promenljivoj
  - primer: 1.2: starost=godine("Petar Petrović"):25
- Argumenti se mogu pisati i sa imenom
  - primer: ime="Petar Petrović"

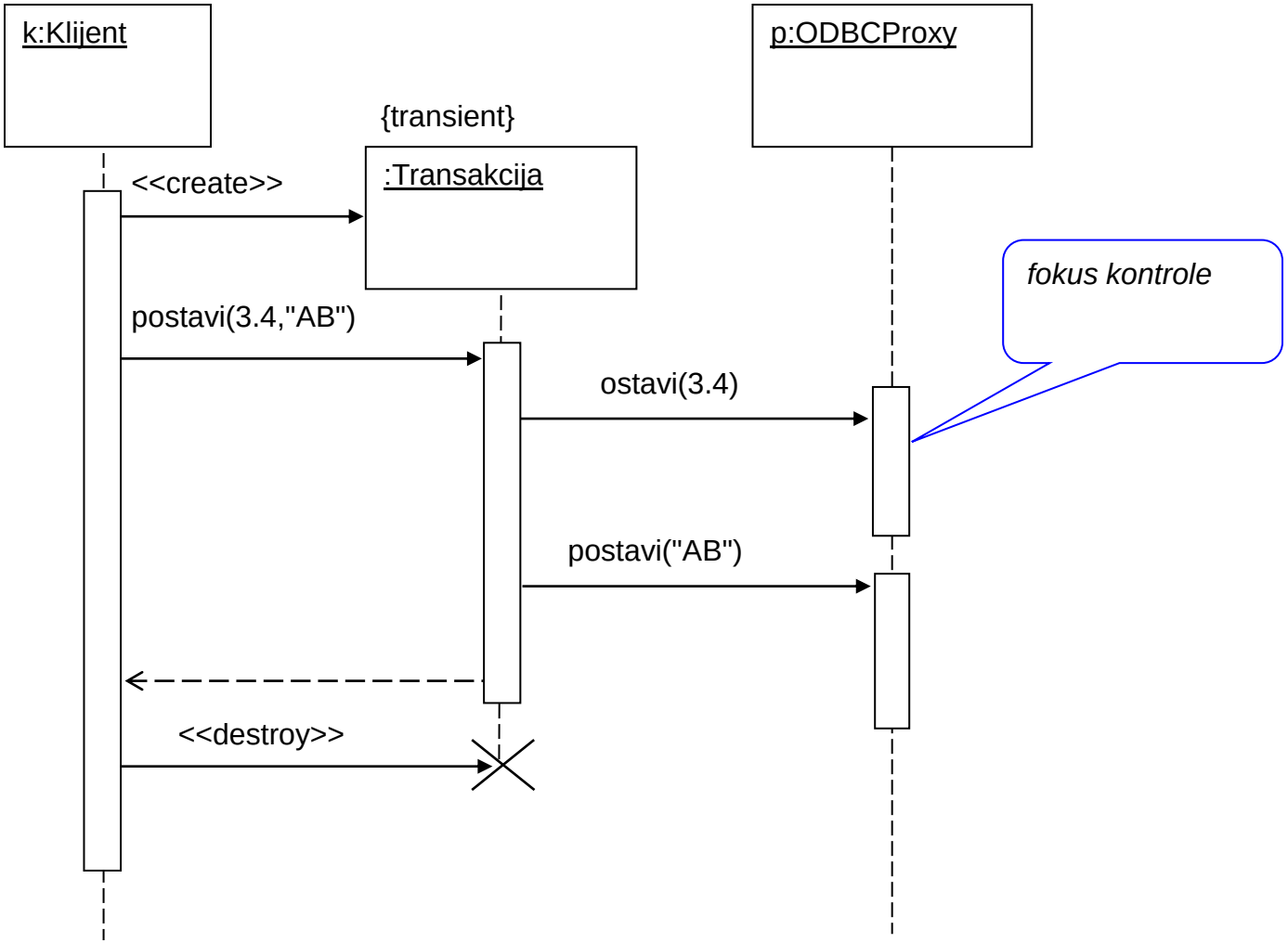
Fokus kontrole (događanje izvršenja)

- Fokus kontrole se može naznačiti samo na dijagramima sekvence
- Fokus kontrole definiše period za vreme kojeg objekat obavlja jednu akciju direktno ili indirektno kroz podređene operacije
- Moguće je i ugnežđivanje fokusa kontrole iz sledećih razloga:
  - (A) zbog rekurzije ili poziva sopstvene (druge) operacije
  - (B) zbog povratnog poziva (*call back*) od pozvanog objekta
- Grafička notacija:





Primer dijagrama sekvence



Vaš zadatak
Šta uraditi?

- Potrebno je napraviti model sa sledećim elementima:
  - Use Case view
  - Logical view
  - Component view
  - Sequence diagram

Vaš zadatak
Use Case view

- Potrebno je imati odgovarajuće Use Case-ove iz SSU-ova za koje se pravi model i odgovarajuće Aktere.
- Na dijagramu Use Case view/main predstaviti te elemente povezane odgovarajućim vezama.

Vaš zadatak
Logical view

- U ovom dijagramu klasa treba predstaviti:
  - serverske Web strane
  - klijentske Web strane
  - php klase koje služe za pristup bazi i njihove odnose; veze tipa:
    - build - serverska strana prva klijentsku, ovo je unidirekciona asocijacija sa stereotipom build
    - link - postoji hiperlink od strane A do strane B, ovo je unidirekciona asocijacija sa stereotipom link
    - submit - kada se pritisne submit dugme na formi A ide se na serversku stranu B, ovo je unidirekciona asocijacija sa stereotipom submit
    - nasleđivanje klasa - Generalize relacija

Vaš zadatak
Component view

- U Component view-u popuniti dijagram komponentama koje predstavljaju fajlove (php) vaše Web aplikacije, a veze između njih odgovaraju require i include direktivama.
- Ako imate mnogo stranica, uvesti pakete po podcelinama.

Deadline ?

- Sve modele je potrebno poslati na SVN/GIT repozitorijum
- Krajnji rok za ovu fazu (subota) 30.05.2015. u 18:00.