

Use Case Diagram

DIJAGRAMI SLUČAJEVA KORIŠĆENJA

Dijagrami slučajeve korišćenja

- ▶ Dijagram slučajeve korišćenja je UML dijagram visokog nivoa koji se koristi za analizu zahteva i ponašanja sistema.
- ▶ Umesto da prikaže detalje pojedinih funkcionalnosti sistema, on se koristi da prikaže same funkcionalnosti koje su dostupne u posmatranom sistemu.
- ▶ Koristi se da grafički modeluje funkcionalnosti sistema u formi slučajeve korišćenja i kako spoljašnji korisnici, koji se nazivaju učesnici, reaguju sa sistemom.
- ▶ Dijagram slučajeve korišćenja ne pokušava da prikaže redosled ili broj koliko puta akcije ili podakcije sistema treba da budu izvršene.

Dijagrami slučajeva korišćenja

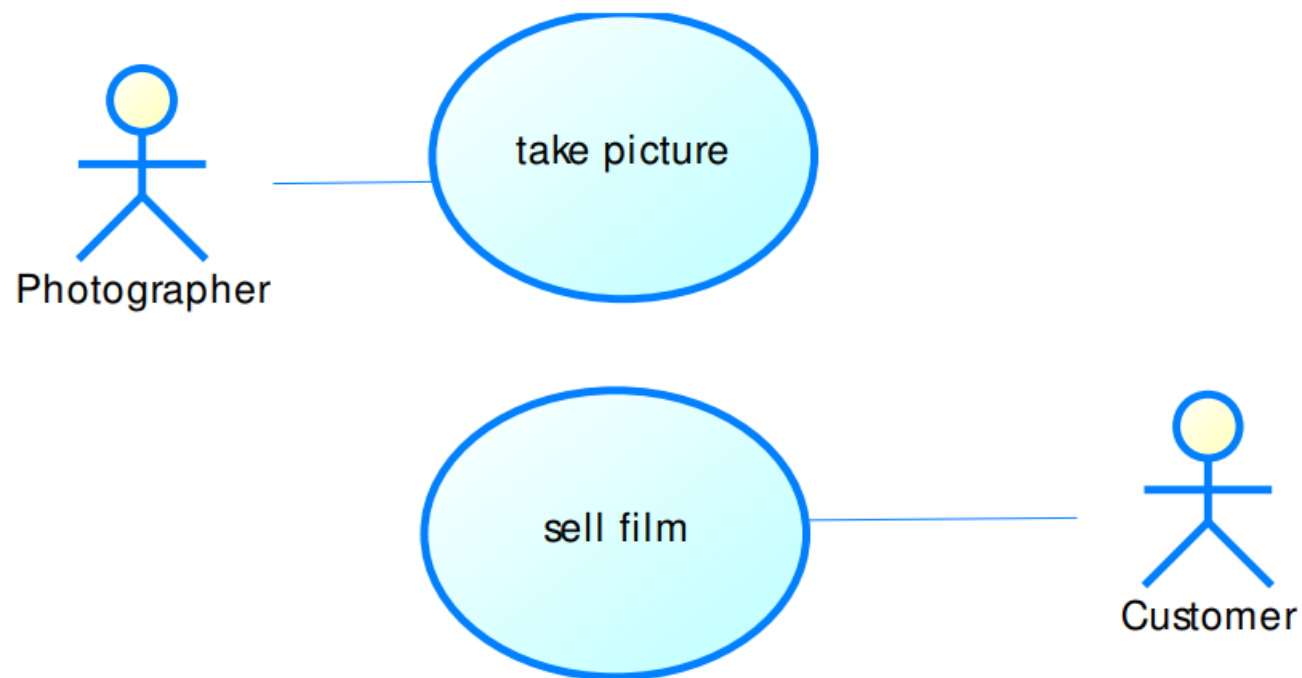
- ▶ Pomoću dijagrama slučajeva korišćenja se odmah vidi prikaz funkcionalnosti sistema.
- ▶ Dodatni detalji se mogu kasnije dodati u dijagram ako je potrebno da se razjasne tačke od interesa u ponašanju sistema.
- ▶ Dijagrami slučajeva korišćenja su pogodni za opisivanje svega što se može uraditi pomoću posmatranog sistema od strane bilo koje osobe koja mu pristupa.
- ▶ Razvoj dijagrama slučajeva korišćenja definiše se sledećim aktivnostima:
 - Definisanjem učesnika
 - Definisanjem slučajeva korišćenja
 - Definisanjem tipova veza između učesnika i slučajeva korišćenja
 - Izradom dijagrama slučajeva korišćenja

Učesnici

- ▶ Učesnik je spoljašnji učesnik ili grupa učesnika koji vrše interakciju sa sistemom.
- ▶ Učesnici mogu biti ljudi ili drugi spoljašnji sistemi. Npr. učesnici u sistemu kompjuterske mreže mogu uključiti administratore sistema, administratore baze podataka i korisnike.
- ▶ Učesnici su obično oni entiteti čije ponašanje ne možemo da kontrolišemo ili menjamo, jer oni nisu deo sistema koji opisujemo.
- ▶ U dijagramu slučajeva korišćenja, učesnik je primarni učesnik za slučaj korišćenja ukoliko zahteva i pokreće izvršavanje akcija koje su prikazane datim slučajem korišćenja.
- ▶ Učesnik je sekundarni učesnik za slučaj korišćenja ako on ne pokreće akciju, nego pomaže slučaju korišćenja da izvrši akciju.

Učesnici

- ▶ Sekundarni učesnik u jednom slučaju korišćenja može biti primarni učesnik u drugom slučaju korišćenja, bilo u istom ili drugom dijagramu.



Definisanje učesnika

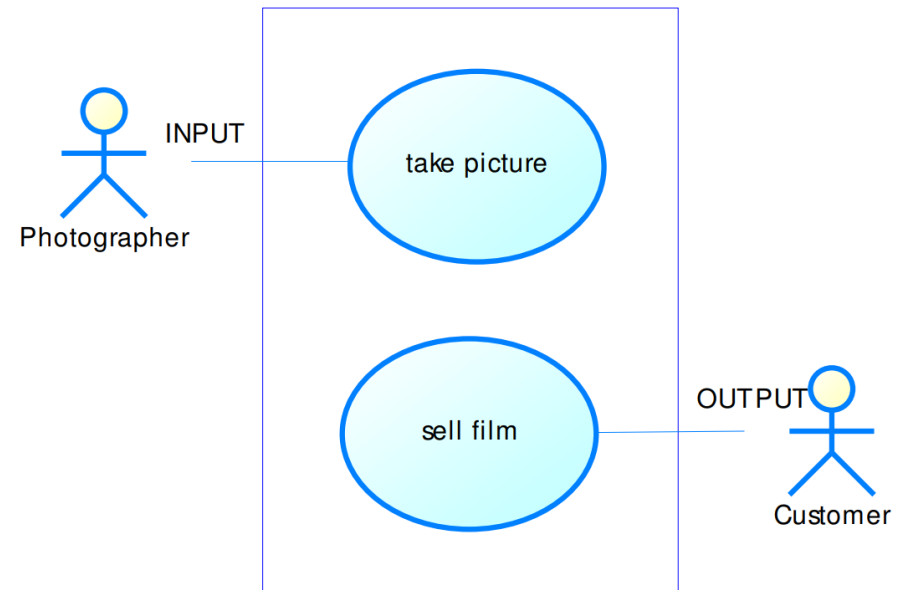
- ▶ Učesnika je moguće identifikovati na osnovu odgovora na sledeća pitanja:
 - Ko će koristiti osnovnu funkcionalnost sistema (primarni učesnici)
 - Ko treba da upravlja, administrira i održava sistem (sekundarni učesnici)
 - Kome će biti potrebna podrška sistema u obavljanju dnevnih zadataka
 - Kojim hardverskim uređajima sistem treba da upravlja
 - Sa kojim drugim sistemima dotični sistem treba da bude u vezi
 - Ko ili šta je zainteresovan za rezultate koje sistem proizvodi

Definisanje veze između učesnika i slučajeva korišćenja

- ▶ Veze koje se uspostavljaju u dijagramu slučajeva korišćenja:
 - Asocijacija (Association)
 - Asocijacija između slučajeva korišćenja tipa <<include>>
 - Asocijacija između slučajeva korišćenja tipa <<extend>>
 - Generalizacija - nasleđivanje (Generalization-Inheritance)
 - Zavisnost (Dependency)

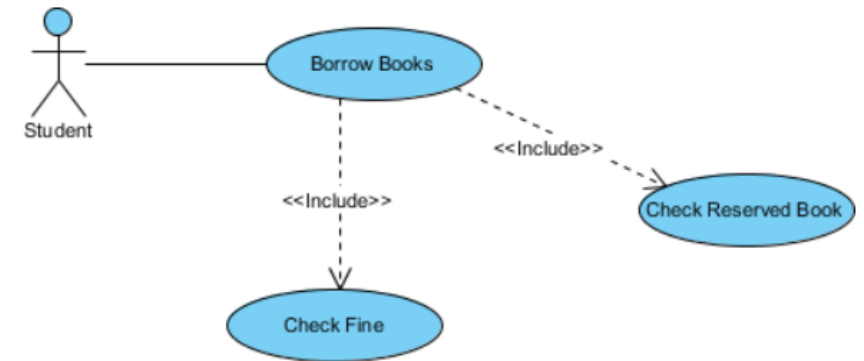
Asocijacija

- ▶ Asocijacija predstavlja vezu između učesnika i slučajeva korišćenja.
- ▶ Orientacija asocijacije je definisana na način kako je ona grafički kreirana.
- ▶ Asocijacija koja vodi od učesnika ka slučaju korišćenja je unutrašnja asocijacija.
- ▶ Asocijacija koja vodi od slučaja korišćenja ka učesniku je spoljašnja asocijacija.



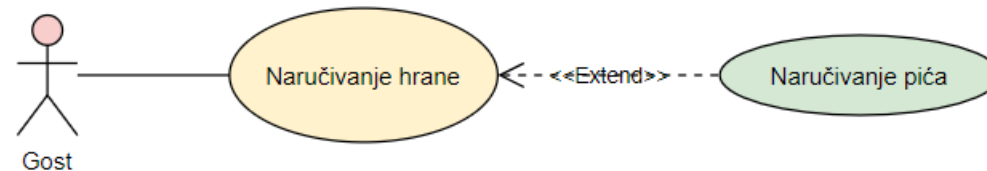
Veza tipa <<include>>

- ▶ Include je direktna veza između dva slučaja korišćenja koja ukazuje da je ponašanje slučaja korišćenja koji se uključuje ubačeno u ponašanje slučaja korišćenja u koji se vrši uključenje. Slučaj korišćenja koji se uključuje može zavisiti od promena nastalih u slučaju korišćenja u koji se vrši uključenje.
- ▶ Vezu include treba koristiti kada postoje zajednički delovi ponašanja dva ili više slučaja korišćenja. Taj zajednički deo se odvaja u poseban slučaj korišćenja kako bi se on uključio u sve bazne slučajeve korišćenja koji imaju taj zajednički deo. Primarna upotreba veze include je ponovna upotreba (eng. reuse) zajedničkih delova u slučajevima kada se bazni slučaj korišćenja ne može koristiti samostalno jer zavisi od delova koji se moraju uključiti.



Veza tipa <<extend>>

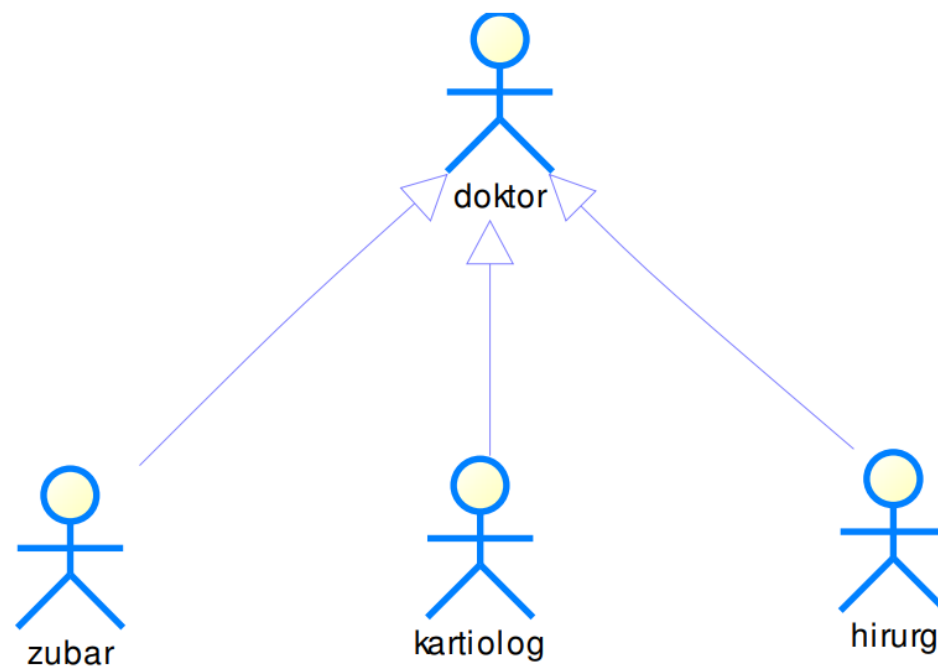
- ▶ Extend je relacija od slučaja korišćenja kojim se vrši proširenje do slučaja korišćenja koji se proširuje kojom se specificira kako i kada se ponašanje definisano u slučaju korišćenja kojim se vrši proširenje može uključiti u slučaj korišćenja koji se proširuje. Proširenje se vrši u jednoj ili više tačaka proširenja u slučaju korišćenja koji se proširuje.
- ▶ Vezu extend treba koristiti kada postoji neko dodatno ponašanje koje treba dodati, moguće uslovno, u ponašanje jednog ili više slučaja korišćenja.



Generalizacija

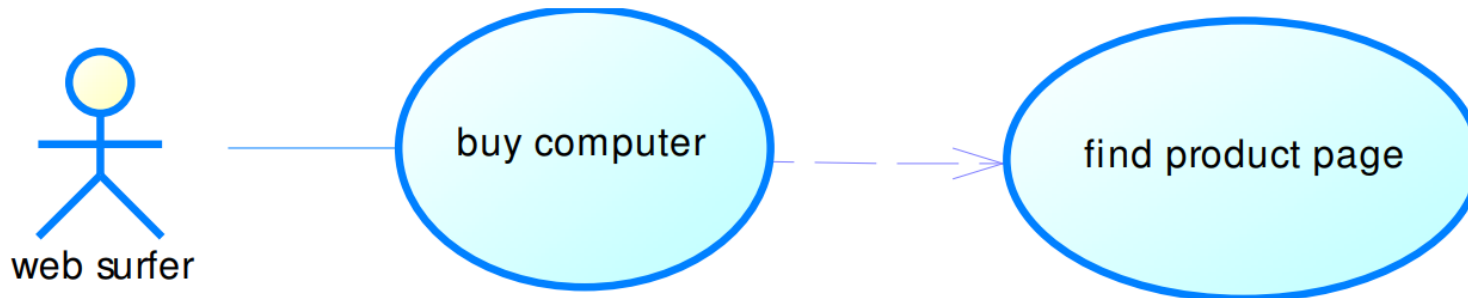
- ▶ Generalizacija (nasleđivanje) je veza u kojoj objekat potomak nasleđuje osobine objekta roditelja.
- ▶ U dijagramima slučajeva korišćenja generalizacija se može kreirati između:
 - dva učesnika
 - dva slučaja korišćenja
- ▶ Npr. dva ili više učesnika mogu imati sličnosti i mogu komunicirati sa istim skupom slučajeva korišćenja na isti način. Ove sličnosti se prikazuju pomoću generalizacije između datih učesnika.
- ▶ U ovom slučaju, učesnici potomci nasleđuju uloge i odnose sa slučajevima korišćenja koji ima učesnik roditelj. Učesnik potomak uključuje attribute i operacije svog roditelja.

Generalizacija



Zavisnost

- ▶ Zavisnost predstavlja vezu koja govori da je jednom objektu za ispunjenje funkcionalnosti potrebna usluga drugog objekta (zavisi od njega). U dijagramima slučajeva korišćenja može se kreirati zavisnost između:
 - učesnika i slučaja korišćenja
 - dva učesnika
 - dva slučaja korišćenja
- ▶ Npr. kupovina kompjutera preko internet sajta uključuje aktivnost pronalaska stranice sa proizvodima u okviru prezentacije prodavca:



Zadatak

- ▶ Napraviti dijagram slučajeve korišćenja za sistem projektnog zadatka.