UNIVERZITET METROPOLITAN BEOGRAD

SE322 - INŽENJERSTVO ZAHTEVA

Modeliranje zahteva

Lekcija 09

PRIRUČNIK ZA STUDENTE

SE322 - INŽENJERSTVO ZAHTEVA

Lekcija 09

MODELIRANJE ZAHTEVA

- Modeliranje zahteva
- Poglavlje 1: Modeliranje zahteva
- Poglavlje 2: Modeli zahteva u slučaju Chemical Traking System
- Poglavlje 3: Vežba
- Poglavlje 4: Domaći zadatak
- Poglavlje 5: Projektni zadatak
- Zaključak

UVOD

Uvodne napomene

Ne može samo jedan pogled na zahteve da obezbedi potpuno razumevanje zahteva. Potrebno je kombinovati tekstualni opis i vizualne opise predstavljanja zahteva i to pri različitim nivoima apstrakcije. Pogledi na zahteve mogu da obuhvate: listu funkcionalnih zahteva, tabele, vizuelne modele analize, prototipove interfejsa korisnika, testove prihvatanja, stabla donošenja odluka, fotografije, video zapise, i matematičke formule. Upoređenjem različitih opise zahteva kreiranih tokom procesa njihovog izazivanja i razvoja, uočavaju se različite nekonzistentnosti, dvosmislenosti, pretpostavke, i ispušteni (neurađeni) delovi izazivanja i analize zahteva, koji se uočavaju primenom samo jednog načina opisivanja zahteva.

Dijagrami efikasnije prikazuju akterima određene tipove informacije nego što to može da se prikaže tekstualno. Slike mogu da mogu da pomognu prevazilaženju jezičkih i terminoloških barijera među članovima projektnog tima. Biznis analitičar može na početku da objasni akterima svrhu korišćenja ovih modela i njihove oznake. Postoji mnogo različitih dijagrama i tehnika modeliranja koje se mogu da izaberu da bi se kreiralo odgovarajuće vizualno predstavljanje zahteva. U ovoj lekciji naučićete nekoliko tehnika za izradu vizualnih modela zahteva. Gledaćemo da u što većoj meri koristimo UML modele, ali ćemo dati i druge koji se u praksi često koriste.

UVODNI VIDEO

Trajanje video snimka: 2min 4sek

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Poglavlje 1

Modeliranje zahteva

VIDEO PREDAVANJE ZA OBJEKAT "MODELIRANJE ZAHTEVA"

Trajanje video snimka: 11min 55sek

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

DIJAGRAMI MODELA ZA OPISIVANJE ZAHTEVA

Nije moguće jednim dijagramom opisati sve zahteve sistema, kao i softverske zahteve. Koriste se različiti dijagrami da bi se istakla posebna svojstva sistema.

Nije moguće jednim dijagramom opisati sve zahteve sistema, kao i softverske zahteve. Koriste se različiti dijagrami da bi se istakle posebna svojstva sistema. Međutim, dijagrami ne mogu da zamene tekstualne opise zahteva. Oni ih samo proširuju da bi uneli veću jasnoću i konkretnost

Vizuelni modeli pomažu da se utvrde nedostajući, neadekvatni i nekonzistentni zahtevi, jer je to teže uvideti pri korišćenju tekstualne dokumentacije. Najčešće se koriste sledeći vizuelni modeli zahteva:

- Dijagrami toka podataka (data flow diagrams -DFDs)
- Dijagrami toka procesa (process flow diagrams),
- UML dijagrami stanja (state diagrams)
- Mape dijaloga (dialog maps)
- Stabla odlučivanja i tabele odlučivanja (decision trees, decision tables),
- Tabele odgovora na događaje (event-response tables)
- Stablo svojstava (feature tree)
- UML dijagrami slučlajeva korišćenja (use case diagrams)
- UML dijagrami aktivnosti (activity diagrams)
- Dijagrami relacija entiteta (entity-relationship diagrams - ERD)
- UML dijagrami klasa (class diagrams)
- UML dijagrami interakcije (UML interaction diagrms)
- UML sequential diagrams (UML sequential diagrams)

Neki od ovih modela već smo izučavali u prethodnim lekcijama (npr. UML slučajevi korišćenja), neke ćemo izučavati i u narednim lekcijama. Isti dijagrami (po nazivu) se koriste i u drugima fazama projektovanja softverskog sistema (npr. UML sekvencijalni dijagrami, UML dijagrami klasa, UML dijagrami stanja, UML dijagrami aktivnosti), međutim, u tim fazama oni koriste mnogo detaljnije informacije koje nastaju tokom procesa razvoja softvera. Mnogi od navedenih dijagrama se menjaju tokom procesa razvoja softvera, jer sa većim znanjem o sistemu koji je u razvoju, i dijagrami se menjaju i po strukturi, a i po sadržaju i detaljnosti informacija. Oni tada postaju **softverski modeli projektnog rešenja sistema** (software design models).

U ovom predmetu (SE322 Inženjerstvo zahteva), mi ćemo koristiti vizuelne modele koji sadrže informacije koje su raspoložive u fazi razvoja zahteva. To su onda **modeli za analizu zahteva**, tj. modeli koncepta. Kada se ovi modeli koriste u analizi zahteva, ovi dijagrami vam omogućuju da opišete problem domena, da kreirate koncepcijske modele novog sistema. Vi možete ove modele da dobijete na osnovu tekstualnih opisa zahteva, koje ste uradili, ili možete ih dobiti direktno, na osnovu razgovora sa korisnicima u fazi izazivanja (elicitation) zahteva.

Primena ovih modela je podržana mnogim softverskim alatima. Njihova primena olakšava iterativan razvoj zahteva, kao i otkrivanje grešaka.

MAPIRANJE REČI KUPCA U KOMPONENTE MODELA ANALIZE

Analitičar dok sluša kupca uočiće ključne reči koje može da prevede u specifične elemente modela

Ako biznis analitičar pažljivo sluša kupca, uočiće neke ključne reči koje može da prevede u specifične elemente modela. Tabela na slici 1 upućuje na moguća mapiranja, tj. povezivanja ključnih reči i komponenata pojedinih modela. Ako za vreme utvrđivanja i razvoja zahteva (izazivanja zahteva), paralelno pišete tekstualna dokumenta o zahtevima i vizuelne modele analize, onda možete lakše da povežete ključne reči iz tekstualnih dokumenata i pojedine elemente određenih vizualnih modela. To vam omogućava povezivanje ovih komponenata modela sa zahtevima korisnika.

U slučaju **Chemical Traking System (CTS)** koristićemo sledeći paragraf o potrebama korisnika koji je šampion proizvoda koji predstavlja klasu Hemičar (Chemist) dostavio biznis analitičaru. Značajne **imenice** su u tekstu boldirane, a *glagoli* sa napisani kosim slovima (italic) a ***kondicionali*** su napisani *kosim i boldiranim slovima**.***

Tip reči	Primeri	Komponente modela analize
Imenica	Ljudi, organiyacije, softverski sistemi, elementi podataka, ili postojeći objekti	• Spoljni entiteti, skladišta podataka, ili tok podataka • Akteri (dijagramislučajeva korišćenja) • Entiteti i njihov atributi • Trake (dijagrami aktivnosti sa trakama uloga) • Objekti sa stanjima
Glagol	Akcije, razmišljanja o tome šta korisnik ili sistemmože da uradi, ili događaji koji mogu da se pojave	• Procesi • Koraci u procesu (dijagram aktivnosti sa trakama uloga) • Slučajevi korišćenja (dijagrami SK) • Relacije (ERD) • Tranzicije (dijagrami stanja) • Aktivnosti (dijagram aktivnosti)
Kondicional	Uslovni logički iskaz, kao što suiskayi tima IF/THEN	• Događaji (tabela događaja i odgovora sistema)Odluke (stablo odlučivanja, tabela odlučivanja, ili dijagram aktivnosti) • Granjanja (dijagram aktivnosti)

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 1.1 Povezivanje reči kupaca i komponenata model analize zahteva

Hemičar ili član osoblja **Skladišta hemikalija** može *podneti* **zahtev *za jednu ili više hemikalija ako je korisnik ovlašćeni podnosilac zahteva***. Zahtev se može i*spuniti* ili *isporukom* **kontejnera hemikalije** koji se već *nalazi* u **zalihama** Skladišta hemikalija ili postavljanjem **naloga** za novi kontejner hemikalije sa spoljnim **prodavcem**. ***Ako je hemikalija opasna***, hemikalija se može *isporučiti* samo ako je korisnik obučen. **Osoba** koja *postavlja* zahtev mora biti u mogućnosti da *pretraži* **kataloge dobavljača** na mreži za određene hemikalije dok *priprema* svoj zahtev. Sistem treba da *prati* **status** svakog hemijskog zahteva od trenutka kada je pripremljen, dok zahtev nije *ispunjen* ili *poništen*. Takođe treba da *prati* **istoriju** svakog hemijskog kontejnera od trenutka kada je *primljen* u **kompaniju** do njegovog potpunog *trošenja* ili *zbrinjavanja*.

IZBOR MODELA

Često razvojni tim mora da koristi sve tipove modela analize. Najčešće izvrši izbor onih koji su odgovarajući za modeliranje najsloženijih i najrizičnijih delova sistema.

Često razvojni tim mora da koristi sve tipove modela analize. Najčešće izvrši izbor onih koji su odgovarajući za modeliranje najsloženijih i najrazličnijih delova sistema, kao i onih delova u kojima se najčešće javljaju dvosmislenosti i neizvesnosti. Dobri kandidati za modelovanje su bezbednosno-kritični sistemi, sigurnosno-kritički sistemi, i ciljno-kritični sistemi jer je uticaj grešaka često vrlo ozbiljan. Takođe, koristite zajedno modele da bi bili sigurni da su kompletni. Na primer istraživanjem podataka u dijagramu objekata, možete otkriti nedostajuće entitete u ERD-u. Na slici 2 dati su saveti o modelima koje treba koristiti ili kombinovati, za određene slučajeve.

Prikazana informacija	Tehnike prezentacije - modeli
Spoljni interfejsi sistema	• **Dijagram konteksta** i dij**agram slučaja** identifikuju objekte van sistema sa kojim su povezani. Dijagram konteksta i **dijagrami toka podataka** pokazuju ulaze i izlaze sistema na visokom nivou apstrakcije. **Mapa eko sistema** utvrđuje moguće sisteme sa kojim je sistem u interakciji, ali uključuje i one sa kojom nije direktno povezan. **Dijagrami aktivnosti sa trakama učesnika** pokazju šta se dešava u interakciji između sistema. • **Detaljni spoljni interfejsa** se zapisuju u ulaznim i izlaznim formatima datoteka ili u izveštajima. Proizvodi koji uključuju komponente i softvera i hardvera obično imaju specifikaciju interfejsa sa definisanim atributima podataka, možda u formi aplikacionog programskog interfejsa (API) ili specifičnih ulaznih i izlaznih signala za hardverske uređaje.
Tok poslovnog proces	• **Dijagram toka podataka** na najvišem nivou apstrakcije pokazuje kako poslovni proces koristi podatke na visokom nivou apstrakcije. **Dijagrami aktivnosti sa trakama učesnika** pokazuju uloge aktera u aktivnostima toka poslovnog **procesa.** • **Detaljniji nivoi dijagrama toka podataka**, ili **dijagrama aktivnosti sa trakama** učesnika predstavljaju tokove procesa sa potrebim detaljima. Slično, **dijagrami stanja** i **dijagrami aktivnosti** se mogu da koriste za visoke i za niske nivoe apstrakcije, mada se najčešće koriste za definisanje detalja procesa.

Slika 1.2 Izbor modela zahteva za slučaj spoljnih iterfejsa sistema i zaslučaj tokova poslovnog procesa

IZBOR MODELA (NASTAVAK)

Preporučeni modeli za podatke, stanja sistema, složenu logiku, korisničke interfejse, opise zadataka korisnika i za nefunkcionalne zahteve.

Prikazana informacija	Tehnike prezentacije - modeli
Definicija podataka i veza među objekatima podataka	• **Dijagram relacija entiteta (ERD)** pokazuje logičke relacije (veze) između objekata podataka (entiteta). **Dijagrami klasa** pokazuju logičke veze između klasa objekata i podatke koji su im pridodati. • **Rečnik podataka** sadrži detaljne definicije struktura podataka i pojedinačne podatke. Složeni objekti podataka sa progresivno razvijaju na njihove konstitutivne elemente.
Stanja sistema i objekta	• **Dijagram stanja** i **tabele stanja** predstavljaju poglede mogućih stanja sistema ili objekta sa visokim nivoom apstrakcije, kao i promene stanja do koji dolazi pod određenim okolnostima. Ovi modeli su korisni kada više slučajeva korišćenja radi sa određenim objektima, te menjaju njihova stanja. • Neki analitičari kreiraju **tabele događaj-odgovor** kao alat okvira, identifikujući spoljne događaje koji pomažu definisanje granice okvira sistema. Možete sa njima da definišete individualne funkcionalne zahteve. Možete takođe da definišete individualne funkcionalne zahteve sa tabelom događaj-odziv detaljišući kako sistem treba da reaguje na kombinaciju spoljnih događaja i stanje sistema. • **Funkcionalni zahtevi** obezbeđuju detalje koji tačno opisuju ponačanja korisnika i sistema koja dovode do promene statusa.
Složena logika	• **Stablo odlučivanja** pokazuje moguće rezultate iz skupa povezanih odluka ili uslova. **Tabela odlučivanja** utvrđuje funkcionalne zahteve povezane sa različitim kombinaciama istinitih i netačnih rezultata za seriju odluka ili uslova.
Korisnički interfejsi	• **Mapa dijaloga** obezbeđuje pogled visokog nivoa apstrakcije predloženog lli sadašnjeg interfejsa, pokazujući različite elemente prikazivanja i moguće puteve navigacije između njih. • **Priče korisnika** i **prototipovi niske tačnosti** pokazuju mapu dijaloga prikazujući sadržaj svakog prikaza na monitoru bez mnogo detalja. Modeli prikaz-akcija-odgovor opisuju sadržaj i zahteve ponašanja svakog od ovih prikaza monitora • **Detaljni prikazi sadržaja monitora** i prototipovi visoke tačnosti pokazuju tačno kako će elementi prikaza da izgledaju. **Definisanje polja podataka** i opisi kontrole korisničkog interfejsa obezbeđuju dodatne detalje.
Opisi zadataka korisnika	• **Priče korisnika, scenariji** i **specifikacije slučajeva korišćenja** opisuju zadatke korisnika na različitim nivoima detaljnosti. • **Dijagrami aktivnosti sa trakama** pokazuju poslovni proces ili interni rad između više aktera i sistema. Dijagrami stanja i dijagrami aktivnosti vizualno prikazuju tok dijaloga i granjanja na alternativne tokove i izuzetke. • **Funkcionalni zahtevi** obezbeđuju detaljne opise interakcije korisnika i sistema radi dobijanja vredanost na kraju. **Slučajevi testiranja** pokazuju alternativni pogled niskog nivoa o tome koje se ponašanje sistema očekuje pod određenim uslovima ulaza, stanja sistema, i akcija,
Nefunkcionalni zahtevi	• **Atributi kvaliteta** i ograničenja obično su napisana prirodnim jezikom, ali im obično nedostaje preciznost i kompletnost.

Slika 1.3 Izbor najprikladnijeg modela zahteva zavisno od prikazanih informacija u modelu

Poglavlje 2

Modeli zahteva u slučaju Chemical Traking System

VIDEO PREDAVANJE ZA OBJEKAT "MODELI ZAHTEVA U SLUČAJU CHEMICAL TRAKING SYSTEM"

Trajanje video snimka: 16min 28sek

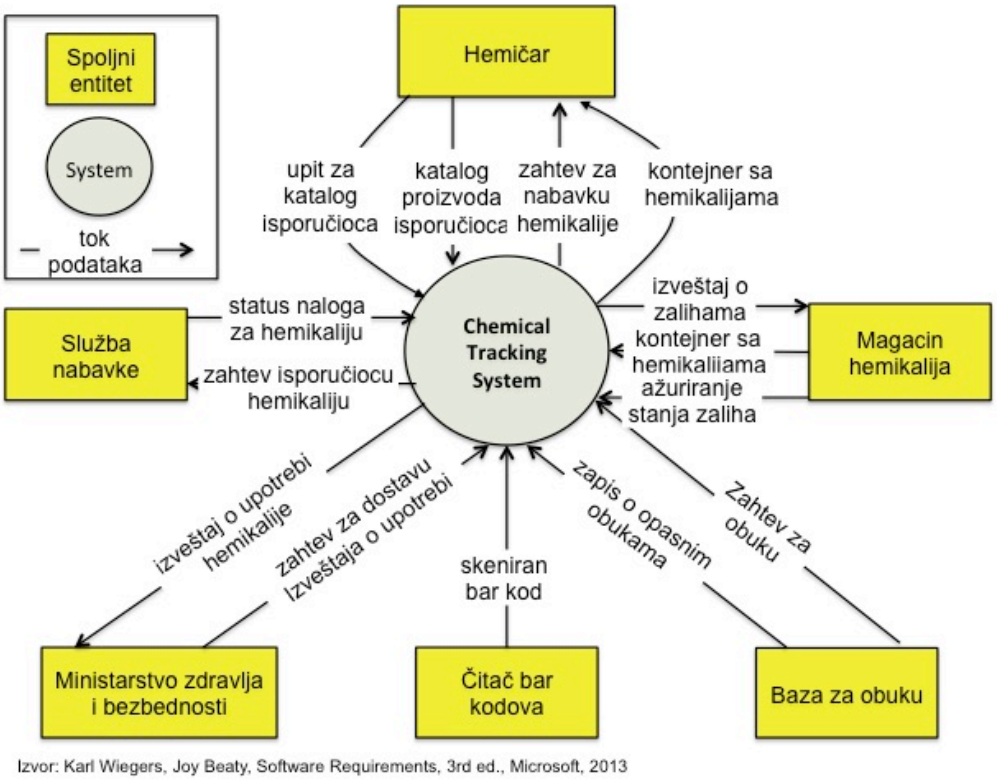
Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

CHEMICAL TRACKING SYSTEM (CTS) - DIJAGRAM KONTEKSTA

Dijagram konteksta utvrđuje tzv. spoljne entitete van granica sistema koji su interfejssa sistemom

Kao što je objašnjeno u lekciji 3, Dijagram konteksta utvrđuje tzv. spoljne entitete (ili terminatore) van granica sistema koji su interfejs (sprega) sa sistemom , kao i podatke, kontrolu, i tok materijala između terminatora i sistema. Slika 1 prikazuje deo dijagrama konteksta za projekat Chemical Tracking System (CTS). Ceo sistem je predstavljen jednim krugom. Očigledno, dijagram konteksta ne daje nikakvu vidljivost unutrašnjosti sistema, koju čine objekti, procesi ili podaci u sistemu

Sada ćemo "ući" u Chemical Tracking System jer ćemo primenom različitih modela koji se koriste za modeliranje zahteva, objasniti kako se realizuje Zahtev za nabavku hemikalije koji inicira Hemičar, tj. spoljni akter sistema.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.1 Deo dijagrama konteksta u projekta ChemicalTracking System

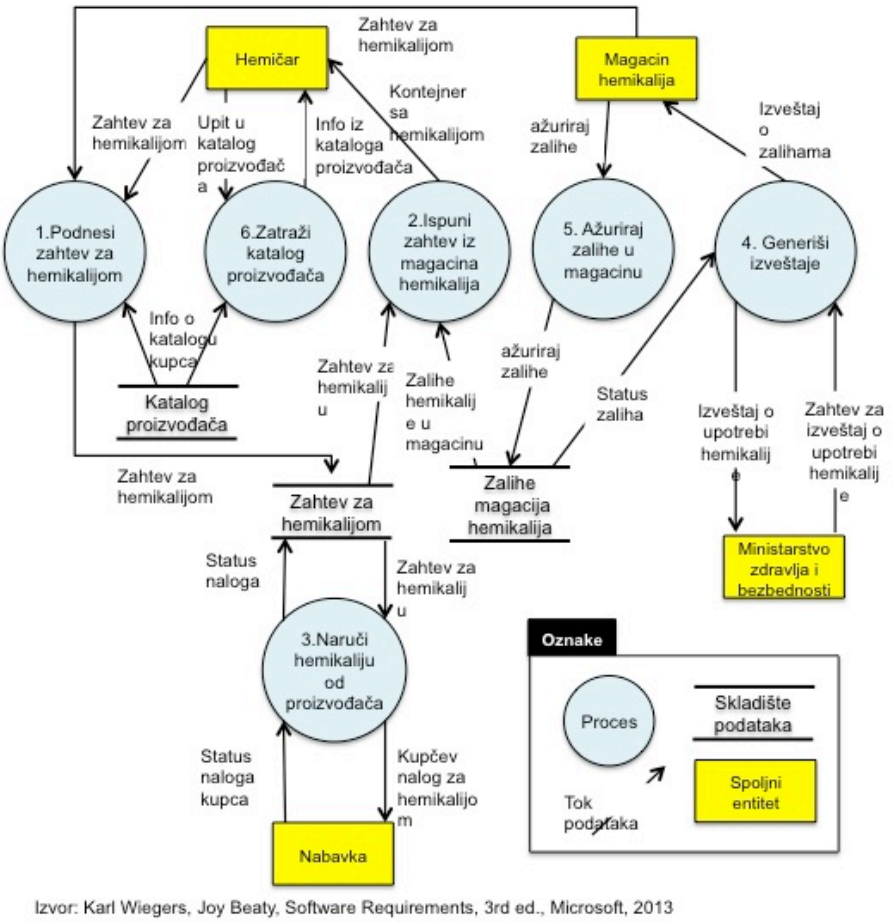
CTS: DIJAGRAM TOKA PODATAKA

Dijagram toka podataka pokazuje kretanje podataka kroz sistem i procese njihove transformacije.

Dijagram toka podataka (Data Flow Diagram - DFD) utvrđuje transformacijske procese sistema, skladištenja podataka ili fizičkih materijala u sistemu sa kojim sistem manipuliše, tokove podataka ili materijala između procesa, skladišta i spoljnjeg okruženja. Modeliranje tokova podataka primenjuje pristup funkcionalne dekompozicije pri analizi sistema, čime razbija složeni problem u sve jednostavnije nivoe sistema. Najviše se koristi kod transakcionih sistema obrade i kod sistema u realnom vremenu.

DFD pokazuje kako se kreću podaci (ili skupa podataka)i kroz sistem, što drugi modeli slabije prikazuju. Sam proces obrade podataka, bolje opisuje slučaj korišćenja ili dijagram aktivnosti. Na slici 2 prikazan je DFD nivoa 0 (najveći stepen apstrakcije), koji deli sistem u više procesa. Na slici se koriste Yourdon-DelMarco oznake. Postoje i malo drugačije oznake u DFD. Svaki od prikazanih 6 procesa, može se prikazati sa više detaljnih procesa na najnižem nivou apstrakcije. Na najnižem nivou se onda procesi prikazuju drugim dijagramima (dijagramom aktivnosti, pseudo kodom, tekstualno...). Funkcionalni zahtevi definišu šta se dešava u svakom od osnovnih (primitivnih) procesa. Svi nivoi DFD moraju da budu međusobno konzistentni. Primenjuju se sledeća pravila:

- Procesi međusobno komuniciraju samo preko skladišta, skladišta komuniciraju međusobno ili sa okruženjem samo preko procesa
- Prikazujte najviše do 8 - 10 procesa u DFD (zbog jasnoće)



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.2 Deo 0 nivoa dijagrama toka podataka u projektu Chemical Tracking System (CTS)

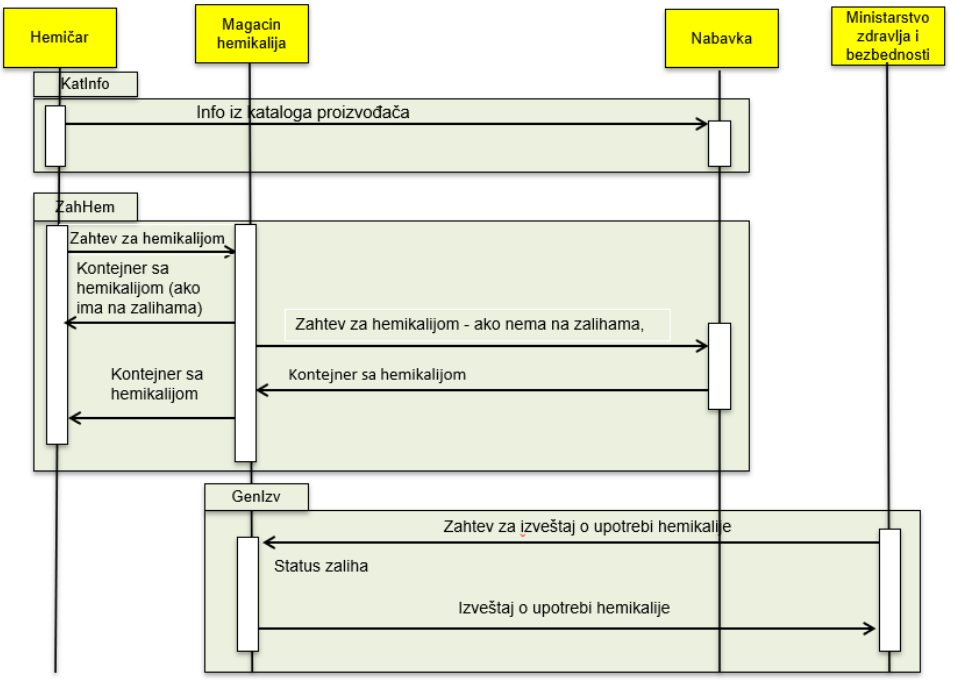
CTS: UML SEKVENCIJALNI DIJAGRAM (1)

U inženjerstvu zahteva, koriste se za grafički opis scenarija prikazanih u slučajevima korišćenja

UML sekvencijalni dijagram (UML Sequential Diagram) prikazuje aktere i entitete sistema, poruke koji oni razmenjuju, duž vertikalnih vremenskih osa na kojima su pravougaonicima obeležena trajanja aktivnosti aktera i entiteta. Oni su vrlo zgodni za prikazivanje redosleda interakcija učesnika u sistemu. Pri razvoju softvera, sekvencijalni dijagrami se koriste u skoro svim fazama razvoja, i sa različitim nivoima apstrakcije. U inženjerstvu zahteva, koriste se za grafički opis scenarija prikazanih u slučajevima korišćenja, a pri projektovanju softvera opisuju interakcija objekata koji realizuju te scenarije. Obično se pri projektovanju koriste više nivoa detaljnosti (odn, apstrakcije) tako da završni nivo daje dovoljno informacija programerima da mogu da uz pomoć odgovarajućeg softverskog alata, da kreiraju odgovarajući kod u nekom od objektno-orijentisanih jezika.

Na slici 3 je prikazana realizacija scenarija zahtevanja određene hemikalije od strane spoljnjeg aktera Hemičar, koji je prethodno prikazan dijagramom toka podataka.

Na dijagramu, umesto 6, prikazano je 3 procesa, jer su neki od procesa prikazanih u dijagramu podataka ovde prikazani kao objedinjeni procesa. Proces ZahHem objedinjuje procese 1, 2, 3 i 5 sa dijagrama toka podataka (slika 2).



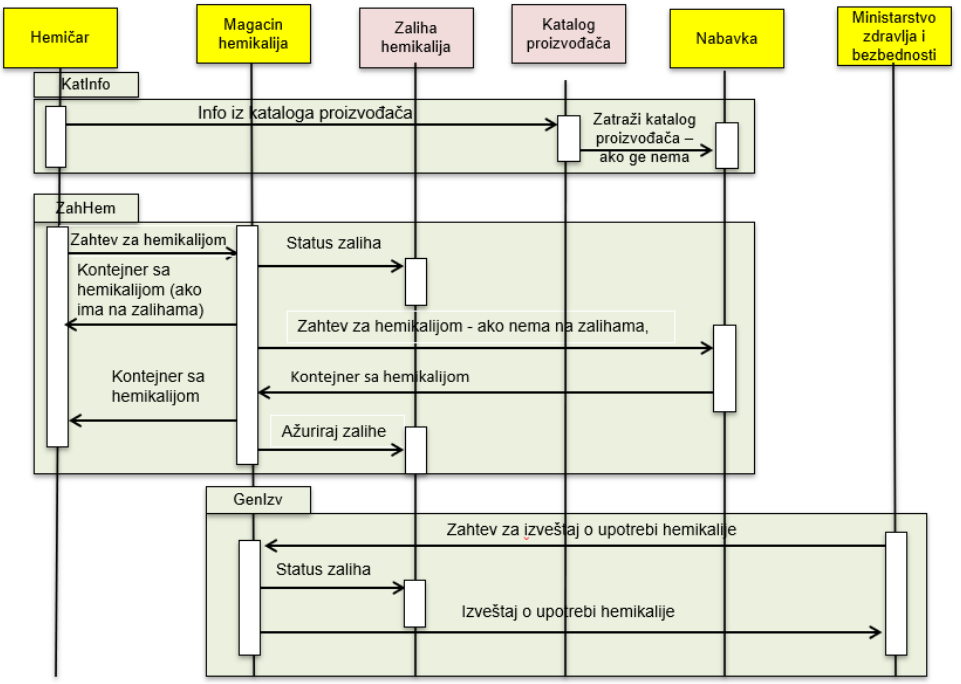
Slika 2.3 Sekvencijalni dijagram scenarija "Zahtev za hemikalijom" sa 4 spoljnja aktera [Wiegers]

CTS: UML SEKVENCIJALNI DIJAGRAM (2)

Pored spoljnih aktera, mogu se dodavati i entiteti sistema.

Slika 4 prikazuje više detalja scenarija sa slika 2, jer uključuje i interne entitete ("Zalihe hemikalija", i "Katalog proizvođača") te je sada scenario bogatiji detaljima, i približniji scenariju prikazanom dijagramom toka podataka (slika 2). Prvi **proces KatInfo** omogućava akteru Hemičar da pogleda katalog proizvoda koji na sistemu održava Služba nabavke, a ako nemaju katalog proizvoda koji se zahteva, onda od proizvođača naručuju bolji katalog. Proces **ZahHem** obezbeđuje kontejner hemikalija akteru Hemičar, ako Magacin hemikalija ga ima na zalihama. Ako ga nema, naručuju kontejnere sa hemikalijom, i isporučuju jedan ili više (prema zahtevu) akteru Hemičar. Ostali kontejneri su sada na Zalihama, i to stanje se ažurira.

Proces **GenIzv** pokazuje kako Magacin hemikalija priprema i šalje izveštaj o upotrebi hemikalije na zahtev Ministarstva zdravlja i bezbednosti, što se po pravilu radi kvartalno. Entitet "Zalihe hemikalija" predstavlja bazu podataka sa svim hemikalijama koje magacin poseduje, sa ažuriranim količinama.



Slika 2.4 Sekvencijalni dijagram scenarija "Zahtev za hemikalijom" sa 4 spoljnja aktera i 2 unutrašnja entiteta [Wiegers]

CTS: TRAKASTI DIJAGRAM PROCESA KUPOVINE HEMIKALIJE

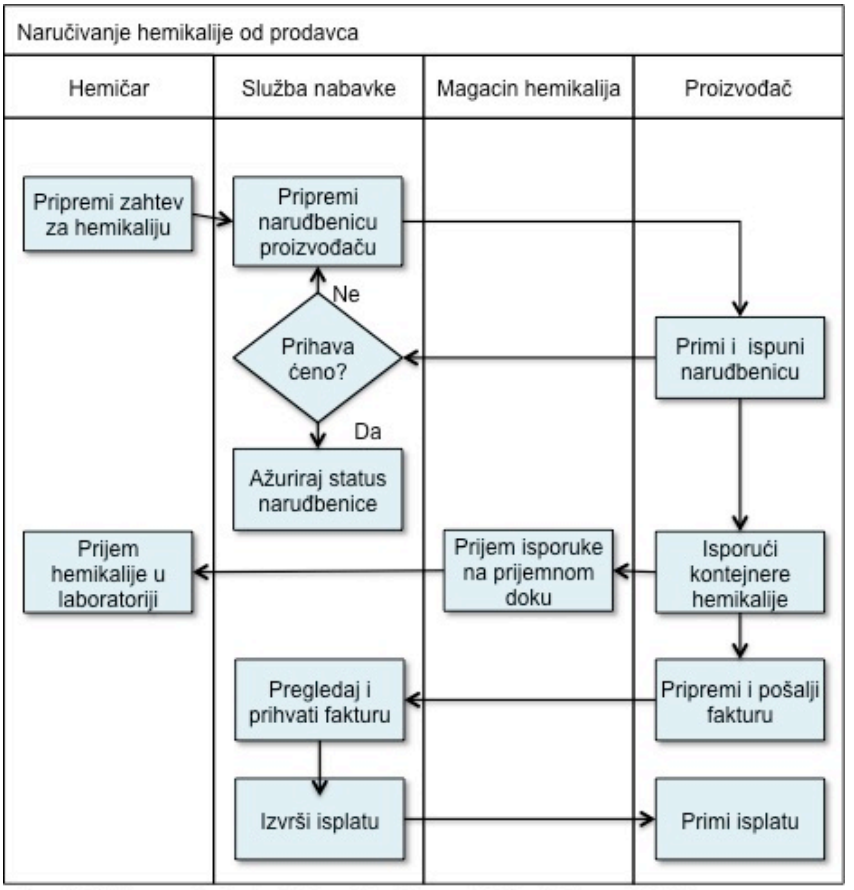
Trakasti dijagrami pokazuju poslovni proces ili operacije predloženog poslovnog procesa.

Trakasti (swimlanes) i UML dijagrami aktivnosti (UML activity diagrams) pokazuju aktivnosti nekog poslovnog procesa, ili operacije predloženog poslovnog procesa. Trakasti dijagram je sličan dijagramu toka, ali je podeljen na tzv. trake, koje predstavljaju ili različite sisteme, ili aktere koji izvršavaju aktivnosti koje se nalaze unutar trake.

Trakasti dijagram pokazuje šta se dešava u jednom procesu pokazan u dijagramu toka podataka (DFD) Oni povezuju funkcionalne zahteve koji omogućavaju korisnicima sistema da obave svoje zadatke. Oni se mogu da iskoriste i za detaljnu analizu radi utvrđivanja zahteva koji podržavaju svaku aktivnost procesa. Crtanje trakastih dijagrama može biti dobar početaka razgovora analitičara sa korisnicima sistema. Trakasti dijagrami mogu da sadrže elemente različitih oblika, ali se najčešće koriste sledeći oblici:

- Aktivnosti, ili koraci procesa u vidu pravougaonika.
- Tranzicije između aktivnosti prikazane su strelama.
- Odluke su prikazane zarotiranim rombom sa ulaznim i izlaznim granama.
- Trake označavaju uloge učesnika, organizacione jedinice, ili sisteme. Mogu se postaviti horizontalno ili vertikalno.

Kod svake aktivnosti razmislite koju funkciju mora da ima sistem da bi ispunio ovu aktivnost. Tako dolazite do funkcionalnih zahteva.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.5 Trakasti dijagram procesa kupovine hemikalije u okviru CTS

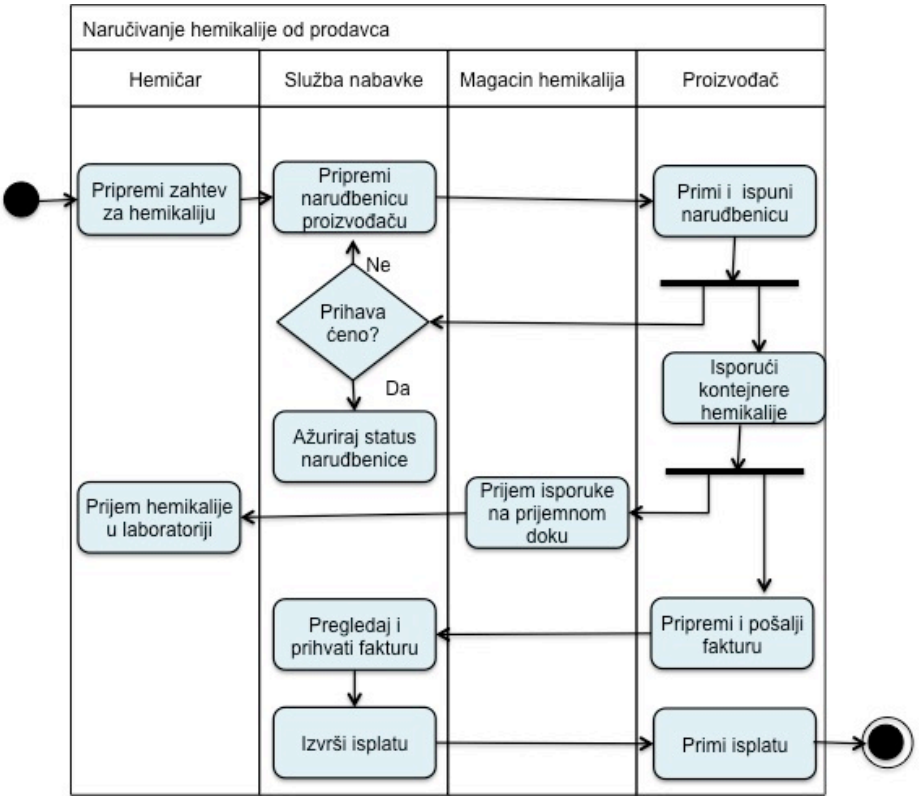
CST: UML DIJAGRAM AKTIVNOSTI ZA PROCES KUPOVINE HEMIKALIJE

UML dijagram aktivnosti je sličan trakastom dijagramu, jer opsiuje poslovni proces.

UML dijagram aktivnosti (UML activity diagram) je sličan trakastom dijagramu, samo su aktivnosti obeležene pravougaonicima sa zaobljenim uglovima, postoje posebni elementi za spajanje i račvanje grana, a postoje i elementi za označavanje početka i kraja procesa. Na slici 6 je prijazan UML dijagram aktivnosti za isti proces nabavke hemikalije, ako što je bio prikazan na slici 5 u slučaju primene trakastog dijagrama.

Kao i u slučaju trakastih dijagrama, i UML dijagrami aktivnosti se mogu podeliti u manje celine da bi se prikazali samo na jednom A4 formatu, koristeći kružne elemente spojnice, koje prespajaju delove dijagrama na različitim listovima. Mogu se koristiti i hijerarhijski nivoi detaljnosti, tako da jedna aktivnost može da predstavlja niz drugih aktivnosti, tj. podskup poslovnog procesa, na nižem, detaljnijem nivou. Primena UML modela se najčešće vrši primenom softverskih alata za izradu modela i i UML dijagrama, kao što je Power Designer koji mi koristimo na vežbama.

UML dijagram aktivnosti, može, kao i UML sekvencijalni dijagram, da se koristi za opisivanje scenarija slučajeva korišćenja i za opisivanje dinamičkog ponašanja sistema. Kao i trakasti dijagram, pogodan je za interakciju sa predstavnicima kupaca ili korisnika, jer je lako razumljiv, te može da posluži i u procesu izazivanja (elicitation) zahteva.



Slika 2.6 Dijagram aktivnosti procesa kupovine hemikalije u okviru Chemical Traktig System [Wiegers]

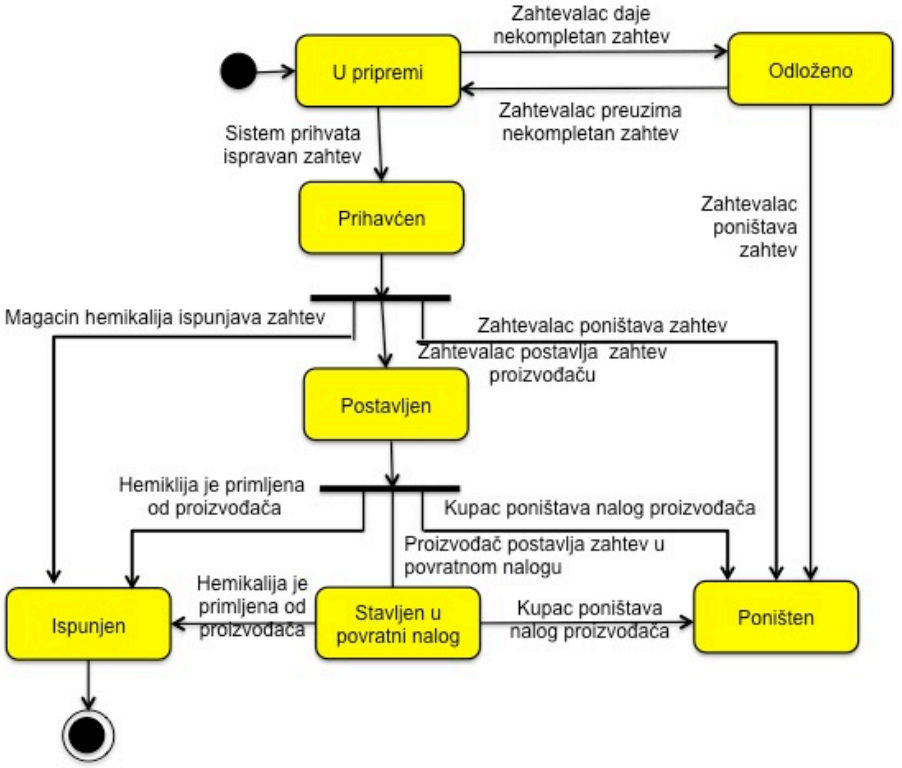
CTS: UML DIJAGRAM STANJA ZAHTEVA ZA HEMIKALIJOM

UML dijagram stanja predstavlja koncizan, kompletan, i nedvosmislen opis stanja nekog objekta u sistemu, kao i uslova za promenu tih stanja

Softverski sistemi imaju kombinaciju funkcionalnog ponašanja, manipulaciju sa podacima i promenu stanaj softverskih objekata koji ga čine, kada se ispune određeni uslovi. UML dijagram stanja (UML state diagram) predstavlja koncizan, kompletan, i nedvosmislen opis stanja nekog objekta u sistemu, kao i uslova za promenu tih stanja (okidači, pokretači promene stanja). Na slici 7 prikazan je UML dijagram stanja zahteva za hemikalijom (to je jedan od softverskih objekata), u kome se vide 7 mogućih stanja zahteva (pravougaonici sa zaobljenim uglovima), uslova za njivo ostvarenje (strelice), kao i početak i kraj dijagrama stanja (crni krug i crni krug upisan u beli krug). Objekat može imati i završna stanja. To su stanja u koje objekat može da dođe, ali koje ne može da promeni (nema izlaznih strelica). UML dijagram stanja na slici 7 prikazuje sledeća završna stanja: "Poništen" i "Ispunjen". Ostala stanja su prelazna stanja. Kao što se vidi, UML dijagram na slici 7 prikazuje životni ciklus jedno zahteva za hemikalijom, koja može biti isporučena zahtevaocu (hemičaru) ako postoji u

magacinu, ili se mora kupiti od proizvođača. Zavisno od uslova, zahtev prolazi kroz navedena stanja, od kojih su dva završna. Stanje "Stavljen u povratni nalog" znači da proizvođač nije u trenutno stanju da ispuni zahtev, is ostavljen je za kasnije.

UML dijagram stanja se ne radi za sve objekte, već samo za one koji su bitni i karakteristični, i čije opisivanje stanja je potrebno programerima pri kodiranju.



Slika 2.7 UML dijagram stanja zahteva za hemikalijom u Chemical Tracking System [Wiegers]

CTS: TABELA STANJA ZAHTEVA ZA HEMIKALIJOM

Tabela stanja prikazuje sve mogće tranzcije između stanja u vidu matrice.

Tabela stanja (state table) prikazuje sve moguće tranzicije (prelaske iz jedno u drugo stanje) između stanja u vidu matrice. Sva moguća stanja se upisuju u prvu kolonu tabele (matrice). Prvi red tabele sadrži takođe sva stanja nekog objekta. Polje u tabeli pokazuje da li stanje navedeno u prvoj koloni može da pređe u stanje dato u prvom redu tabele, i pod kojim uslovima.

Na slici 8 prikazana je tabela stanja koja odgovara UML dijagramu stanja datom na slici 7. Dijagram i tabela sadrže potpuno iste informacije, međutim, tabela nam omogućava da proverimo da li možda nismo ispustili neku tranziciju, a dijagram stanja olakšava razumevanja stanja i prelazaka stanja nekog objekta jer je vizuelno predstavljeno, što je obično lakše za razumevanje. Završna stanja imaju samo "ne" u svom redu, jer nema izlaska objekta iz tok stanja.

Testeri mogu na osnovu dijagrama stanja i tabela stanja da naprave testove, što može da doprinese ranom otkrivanja eventualnih grešaka u radnoj verziji sistema.

	U pripremi	Odloženo	Prihvaćeno	Postavljeno	U povratnom nalogu	Ispunjen	Poništen
U pripremi	ne	Zahtevalac daje nekompletan zahtev	Sistem prihvata ispravan zahtev	ne	ne	ne	ne
Odloženo	Zahtevalac preuzima nekompletan zahtev	ne	ne	ne	ne	ne	ne
Prihvaćeno	ne	ne	ne	Zahtevalac postavlja zahtev proizvođaču	ne	Magacin hemikalija ispunjava zahtev	Zahtevalac poništava zahtev
Postavljeno	ne	ne	ne	ne	Proizvođač postavlja zahtev u povratnom nalogu	Hemiklija je primljena od proizvođača	Kupac poništava nalog proizvođača
U povrtnom nalogu	ne	ne	ne	ne	ne	Hemikalija je primljena od proizvođača	Kupac poništava nalog proizvođača
Ispunjen	ne	ne	ne	ne	ne	ne	ne
Poništen	ne	ne	ne	ne	ne	ne	ne

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.8 Tabela stanja zahteva za hemikalijom u projektu Chemical Tracking System

CTS: MAPA DIJALOGA ZA SLUČAJ ZAHTEVA ZA HEMIKALIJOM

Mapa dijaloga pokazuje elemente dijaloga i navigacione veze između njih.

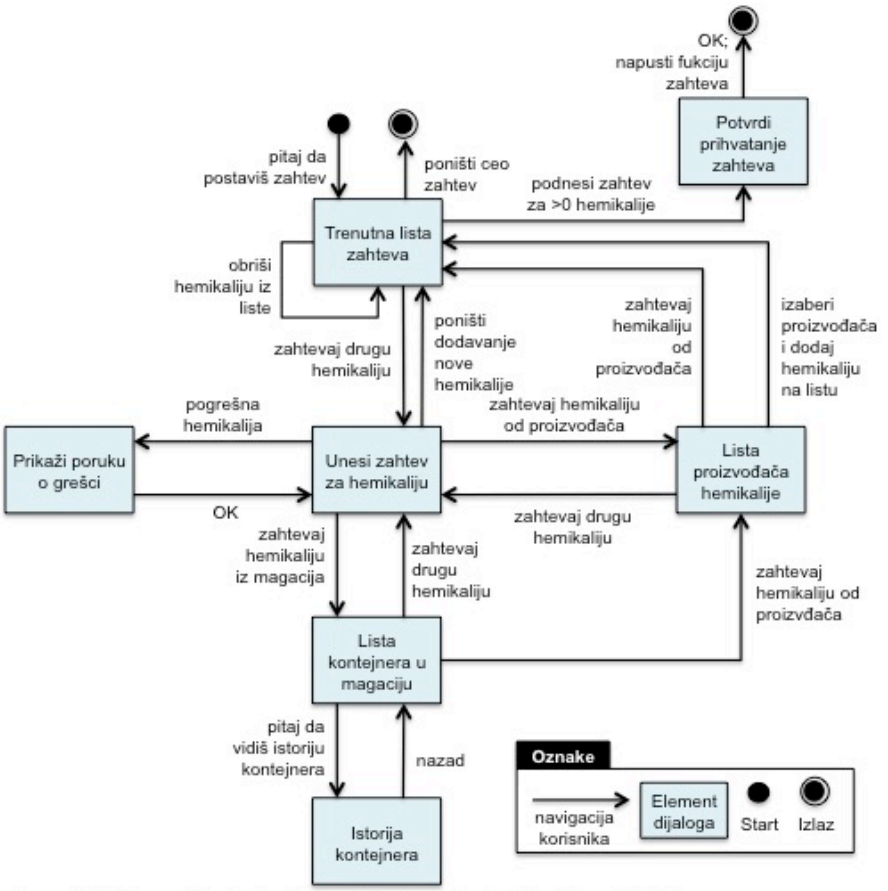
Mapa dijaloga (dialog map) predstavlja projektovanje korisničkog interfejsa na visokom nivou apstrakcije. Ona pokazuje elemente dijaloga i navigacione veze između njih, ali ne pokazuje detaljna dizajnerska rešenja korisničkog interfejsa. U skladu sa navigacionim vezama, korisnik bira elemente dijaloga.

Mapa dijaloga vam omogućava da istražujete koncepte različitih hipotetički korisničkih interfejsa. Programer i korisnik preko mape dijaloga dogovaraju korisnčki interfejs i način interakcije. Dijalog mapa može biti korisna i za modeliranje vizualne arhitekture nekog veb sajta. Dijalog mapa se vezuje za priču korisnika i uključuje i kratak opis svrhe svake interakcije.

Korisnici mogu da prate tokove interakcije i da kažu nedostajaće, netačne ili nepotrebne zahteve. Konceptualna dijalog mapa koja se koristi pri analizi zahteva, služi kao vodič za vreme detaljnog projektovanja korisničkog interfejsa.

Navigacija korisnika zavisi od postavljenih uslova, akcije korisnika, vrednosti nekog podatka, i kombinacije ovih faktora. Dijalog mapa je idealni način za predstavljanje interakcije aktera i sistema u slučajevima korišćenja. Može prikazati alternativne scenarije i grane normalnog toka. Mapa dijaloga na slici 9 prikazuje interakcije korisnika i sistema u slučaju korišćenja u

kom hemičar zahteva hemikaliju. Tu se koristi normalni (isporuka iz magacina) i alternativni scenario (kupovina hemikalije).



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.9 Mapa dijaloga zahteva za hemikalju u Chemical Tracking System

CTS: TABELA ODLUČIVANJA U ZAHTEVU ZA NOVOM HEMIKALJOM

Tabela odlučivanja opisuje šta sistem treba da radi i koje odluke da donese u određenim složenim uslovima.

Softverski sistem često koriste složenu logiku u radu, sa različitim kombinacijama uslova, što dovodi do različitog ponašanja sistema.

Tabela odlučivanja (decision table) opisuje šta sistem treba da radi i koje odluke da donese u određenim složenim uslovima. Tabela odlučivanja sadrži listu različitih vrednosti za sve faktore koji utiču na ponašanje sistema, i očekivane akcije sistema kao odgovor na takve uslove.

Slika 10 pokazuje tabelu odlučivanja koja pokazuje kada će Chemical Tracking System da prihvati ili da odbije zahtev za novu hemikaliju. Navedeno je 4 uslova, koja mogu ili da budu tačni ili pogrešni (to su logičke vrednosti), te ima 2^4, tj. 16 mogućih kombinacija. tj. funkcionalnih zahteva. U praksi se neke od tih kombinacija isključuju, te je manje stvarna.

Slika takođe navodi jednu dve moguće akcije u slučajevim ispunjenja navedenih 5 realnih kombinacija. To su akcije prihvatanja ili odbijanja zahteva.

Broj zahteva:					
Uslov	**1**	**2**	**3**	**4**	**5**
Koisnik je ovlaščen	F	T	T	T	T
Hemikalija je raspoloživa	-	F	T	T	T
Hemikalija je opsana	-	-	F	T	T
Zahtevaoc je obučen	-	-	-	F	T
Akcija					
Zahtev je prihvaćen			X		X
Zahtev je odbijen	X	X		X	

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

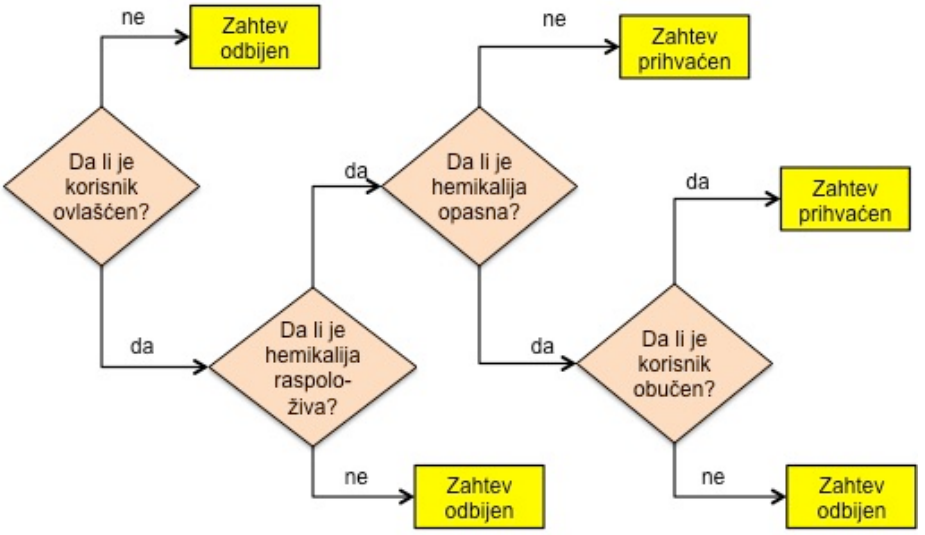
Slika 2.10 Tabela odlučivanja u slučaju zahteva za hemikalijom u Chemical Tracking System

CTS: STABLO ODLUČIVANJA U SLUČAJU ZAHTEVA ZA HEMIKALIJOM

Stablo odlučivanja dokumentuje grafičkim putem poslovna pravila koje treba ugraditi u softver.

Na slici 11 prikazano je stablo odlučivanja koja sadrži istu logiku kao i prikazana tabela odlučivanja, jer se odnosi na isti primer. Pet žutih pravougaonika pokazuju moguće rezultate primenjene logike, koja je sadržana u izokrenutim rombovima i koji sadrže logičke iskaze.

I tabele odlučivanja i stabla odlučivanja su zgodni načini dokumentovanja zahteva, tj. poslovnih pravila, što je potrebno da bi se proverilo da li neka kombinacija odlučivanja nije uvrštena u zahteve, tj. u poslovna pravila. Tabela i stablo odlučivanja su lake za tumačenje, za razliku od teksta, koji pri složenim uslovima, nije lako protumačiti.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.11 Stablo odlučivanja za slučaj zahteva za isporuku hemikalije u Chemical Tracking System.

TABELE DOGAĐAJA I ODZIVA SISTEMA

Tabela događaja i odziva sistema navodi sve događaje i očekivano ponašanje sistema na takve događaje.

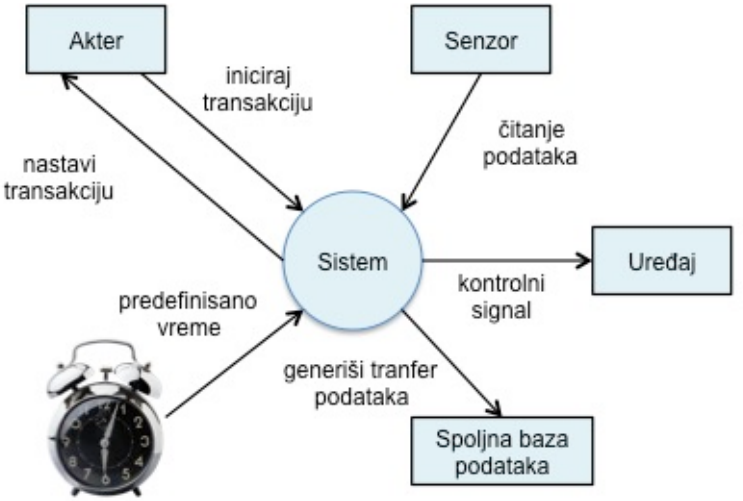
Slučajevi korišćenja i priče korisnika nisu uvek dovoljni za otkrivanje funkcionalnosti koje programeri treba da ostvare. Ovo je naročito karakteristično za sisteme u realnom vremenu. Na primer, sistem za upravljanje semaforima, kamerama i dr. na većim raskrsnicama.

Drugi način za utvrđivanje i predstavljanje zahteva korisnika je da se utvrde spoljni događaji (events) na koje sistem mora da odgovori. Događaj je promena ili aktivnost koja se javi u okruženju korisnika, a koja stimuliše odgovor sistema. Na primer, u slučaju požara (događaj), senzor za registrovanje dima se aktivira i šalje signale ostalom delu sistema čime javlja postojanje požara na lokaciji gde se nalazi.

Tabela događaja i odziva sistema (koriste se i termini: tabela događaja, ili lista događaja) navodi sve takve događaje i očekivano ponašanje sistema na takve događaje. Postoje tri klase događaja sistema:

- poslovni događaj (business event) je akcija čoveka koja stimuliše dijalog sa softverom, što je slučaj aktiviranja nekog slučaja korišćenja.
- događaj signala (signal event) se registruje kada sistem dobije neki kontrolni signal, čitanje podataka, ili prekid rada od strane nekog spoljnjeg hardverskog uređaja, ili nekog drugog sistema.
- vremenski događaj (time event) se aktivira kada sat računar pokaže definisano vreme.

Na slici 12 naveden je primer sa sve tri vrste događaja,



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.12 Odgovor sistema na poslovne, kontrolne i vremenske događaje

PRIMER TABELE DOGAĐAJA I ODZIVA SISTEMA

Tabele događaja i odziva se najviše koriste kod sistema u realnom vremenu.

Tabele događaja i odziva se najviše koriste kod sistema u realnom vremenu. Da bi utvrdili događaje, uzmite u razmatranje sva stanja objekta koji analizirate, i utvrdite događaje koji mogu da objekat dovedu u ta stanja. Pogledajte dijagrame konteksta da bi videli da li ima spoljnih sistema koji bi mogli da iniciraju neki događaj ili da zahtevaju automatski odgovor.

Na slici 13 je data tabela događaja i odziva sistema za upravljanje radom automobilskih brisača. Sem br. 6 (vremenski događaj), svi ostali događaji su tipa kontrolnog signala. Možete da uočite da odgovor sistema ne zavisi samo od događaje, već i od stanja u kome se sistem nalazio u momentu javljanja događaja. na primer, događaji 4 i 5 različite odzive sistema zbog razlike početnog stanja sistema.

U tabele događaj i odziva sistema, zavisno od oblasti primene, mogu se dodavati:

- frekvencije događaja ili broj javljanja događaja
- elementi podataka koji su potrebni za obradu događaja,
- stanje sistema posle dejstva i obrade događaja.

Cilj je da se obezbedi dovoljno informacija za programera i testera da oni mogu da odrade svoj posao. Table događaja i odziva ne daje nikakvu informaciju o načinu primene. To je posao projektanta sistema, a programera da kodira to projektno rešenje. a testera da pripremi testove.

ID	Događaj	Stanje sistema	Odgovor sistema
1	Postavi kontrolu brisača na malu brzinu	Brisač isključen, uključena velika brzina ili povremeno brisanje	Postavi brznu mootora brisača na malu brzinu
2	Postavi kontrolu brisača na veliku brzinu	Brisač je isključen, na niskoj brzini, ili na povremeno brisanje	Postavi brznu mootora brisača na veliku brzinu
3	Isključi brisače	Brisač je na velikoj brzini, maloj brzini ili u povremenom radu	1. Završi postojeći ciklus rada brisača 2. Isključi motor brisača
4	Uključi brisače na povremeno brisanje	Brisač je isključen	1. Izvrši jedan cikljus brisanja 2. Učitaj vremenski interval rada brisača 3. Inicijalizuj tajmer brisača
5	Uključi brisače na povremeno brisanje	Brisač je na maloj ili na velikoj brzini	1. Završi započeti ciklus 2. Učitaj vreme intervala rada 3. Inicijalizuj tajmer brisača
6	Definiše vreme rada brisača od poslednjeg brisanja	Brisač je u povremenom radu	Izvrši ciklus brisača sa malom brzinom.
7	Promeni interval povremenog brisanja	Brisač je u povremenom radu	1. Učitaj interval vremena rada brisača 2. Inicijalizuj tajmer brisača
8	Promeni interval povremenog brisanja	Brisač je isključen, na visokoj brzini, ili na maloj brzini	Nema odgovora
9	Projem neposrednog signala od brisača	Brisač je isključen	Izvrši jedan ciklus sa malom brzinom.

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.13 Tabela događaja i odziva sistema za slučaj sistema za upravljanje radom brisača automobila

PRIMENA MODELA PRI AGILNOM RAZVOJU SOFTVERA

Razlika tradicionalnih i agilnih metoda razvoja pri modeliranju zahteva se odnosi samo na to kada se kreira neki model i nivo detalja koji on sadrži.

Svi projekti treba da istražuju zahteve, da postave modele zahteva, i da analiziraju zahteve iz različitih perspektiva, bez obzira na primenjen postupak razvoja. Izbor modela je vrlo sličan. Razlika u tome kako tradicionalni i agilni metodi razvoja vrše modeliranje se odnosi samo na to kada se kreira neki model i nivo detalja koji on sadrži.

Na primer, na početku agilnog projekta možete pripremiti DFD model nivoa 0, a kasnije, u iteraciji, da definišete detalje, zavisno od postavljenog cilja iteracije.

Pri agilnom razvoju, možete kreirati modele koji su u manjem perfektnom formatu, u vidu neke skice na tabli, i da ih ne memorišete u dokumentaciji. Kako se primenjuju priče korisnika, i modeli se ažuriraju i dopunjuju.

Ključna stvar i kod agilnih i kod tradicionalnih metoda razvoja je da da se usmerite samo na projekte koji su vam potrebni, i radite ih samo kada su vam potrebni, i radite ih na nivou

detalja koliko je to potrebno. Ako radite u projektu koji primenjuje agilne metode razvoja softvera, nemojte da odbacite unapred nijedan model za opisivanje zahteva.

VIDEO 17 - ALTERNATIVE REQUIREMENTS VIEWS - WIEGERS (VIDEO)

Trajanje: 5:53 minuta

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

VIDEO 20 - THE DIALOG MAP - WIEGERS (VIDEO)

Trajanje: 7:49 minuta

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

VIDEO 21 - DECISION TABLES AND DECISION TREES - WIEGERS (VIDEO)

Trajanje: 4:38 minuta

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

VIDEO 29 - HOW DETAILED SHOULD REQUIREMENTS BE - WIEGERS (VIDEO)

Trajanje: 7:50 minuta

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

Poglavlje 3

Vežba

MODELI ZA VIZUELNO PREDSTAVLJANJE ZAHTEVA

Mnogi od modela za vizuelno predstavljanje zahteva su jako čitljivi, čak i za stejkholdere koji nisu potkovani tehničkim veštinama, a moraju biti uključeni

Modeli za vizuelno predstavljanje zahteva mogu da budu jako korisni prilikom utvrđivanja i provere zahteva. Mnogi od njih su jako čitljivi, čak i za stejkholdere koji nisu potkovani tehničkim veštinama za proces inženjerstva zahteva. S druge strane, cilj modela je da pomogne prilikom projektovanja sistema koje tek treba da usledi. Na početku će vam pomoći da identifikujete zahteve, zatim da ih dekomponujete ili nadogradite. Oni su jako važni jer, uz njihovo korišćenje, imaćete gotove ulaze za fazu projektovanja i arhitekture, a kasnije i implementacije. Postavljanje zahteva je jako bitno i za fazu testiranja, jer kada imate definisano šta se očekuje od sistema, jasno je šta treba proveriti u njegovom radu i na koji način.

Na vežbama iz ovog predmeta je stavljen **akcenat na sledeće modele za vizuelizaciju zahteva**:

- Dijagram toka podataka
- Dijagram aktivnosti
- Dijagram stanja
- Sekvencijalni dijagram
- Dijagram slučajeva korišćenja

MODEL TOKA PODATAKA

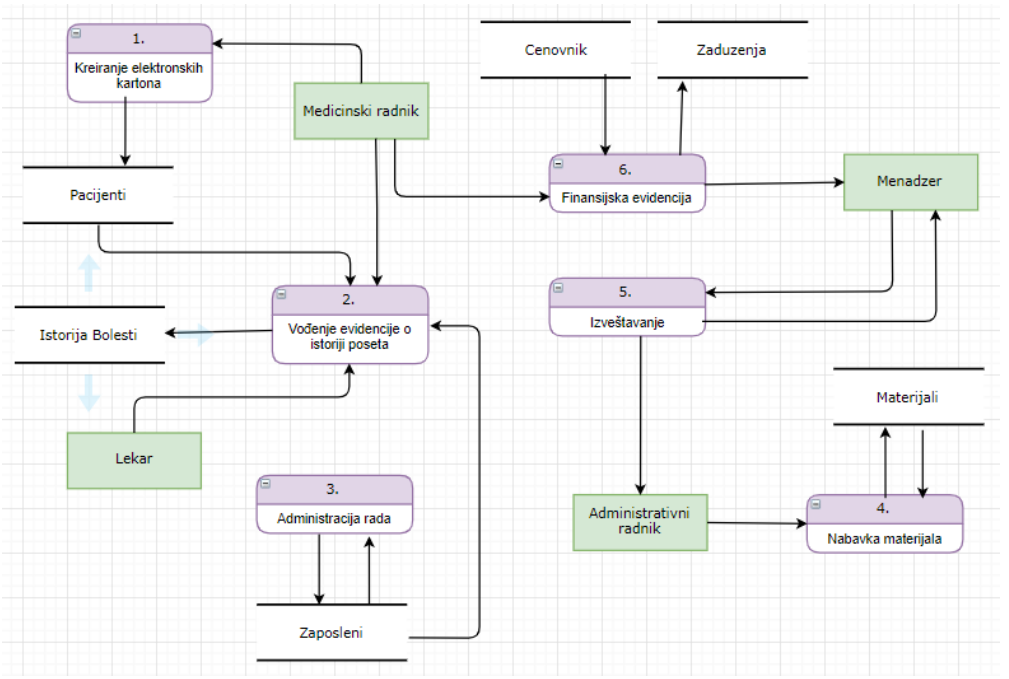
Elementi dijagrama toka podataka su: eksterni entiteti, tokovi podataka, skladišta podataka i procesi.

Model toka podataka se bazira na ideji da se sistem može modelirati kao vizualizacija interakcije podataka koju ima čitav sistem ili neki od njegovih delova sa drugim internim ili eksternim aktivnostima.

Elementi dijagrama toka podataka su: eksterni entiteti, tokovi podataka, skladišta podataka i procesi.

Pri crtanju dijagrama tokova podataka (DTD) se polazi od **dijagrama konteksta** koji se funkcionalnom dekompozicijom dekomponuje na dijagrame nižeg nivoa. Dijagram konteksta

se najpre dekomponuje na **dijagram toka nultog nivoa**, koji je prikazan na slici 1. Najniži nivo dijagrama toka podataka se zove **primitivni dijagram toka podataka**.
Metoda modeliranja zahteva dijagramom toka podataka je poznata kao jedna od strukturnih metoda za modelovanje zahteva. Pored ovih, ističu se objektno-orijentisane metode za modelovanje podataka.



Slika 3.1 Dijagram toka podataka za sistem privatne klinike [Izvor: Marina Damnjanović]

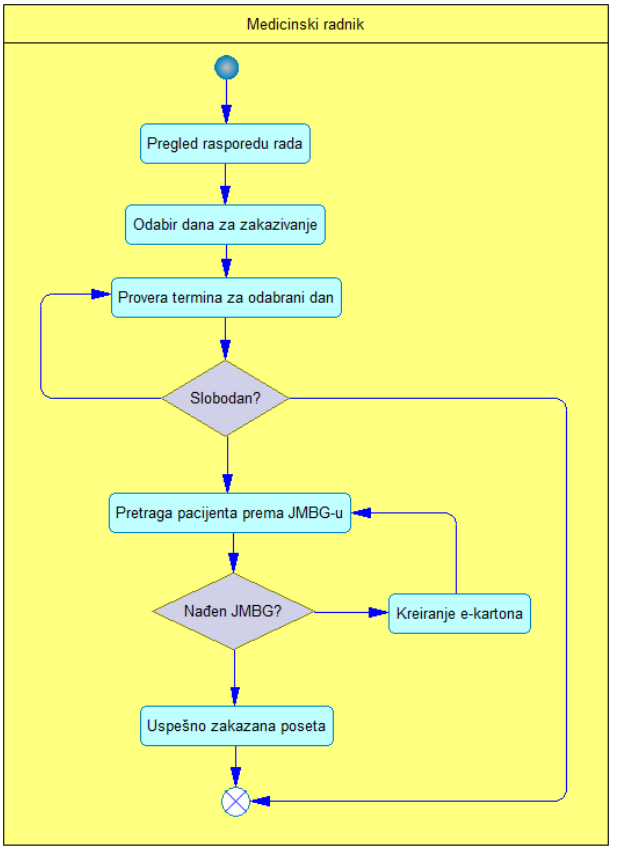
DIJAGRAM AKTIVNOSTI

Glavni elementi dijagrama aktivnosti su: aktivnost, tok, inicijalno stanje, krajnje stanje, tačka odlučivanja, tačka sinhronizacije i organizaciona jedinica (swimline).

Dijagram aktivnosti je važan UML dijagram ponašanja koji opisuje dinamičke aspekte sistema. Dijagram aktivnosti je u osnovi napredna verzija dijagrama toka koja modelira protok iz jedne aktivnosti u drugu aktivnost.

Dijagram aktivnosti često opisuje kako su događaji u određenom slučaju upotrebe povezani, da li se aktivnosti slučaja upotrebe preklapaju i zahtevaju dodatnu koordinaciju. Za identifikovane slučajeve upotrebe koji zahtevaju opisivanje pomoću dijagrama aktivnosti, treba uzeti u obzir preduslove i post uslove i detaljno modelirati složene operacije i aktivnosti.

Glavni elementi dijagrama aktivnosti su: aktivnost, tok, inicijalno stanje, krajnje stanje, tačka odlučivanja, tačka sinhronizacije i organizaciona jedinica (swimline).

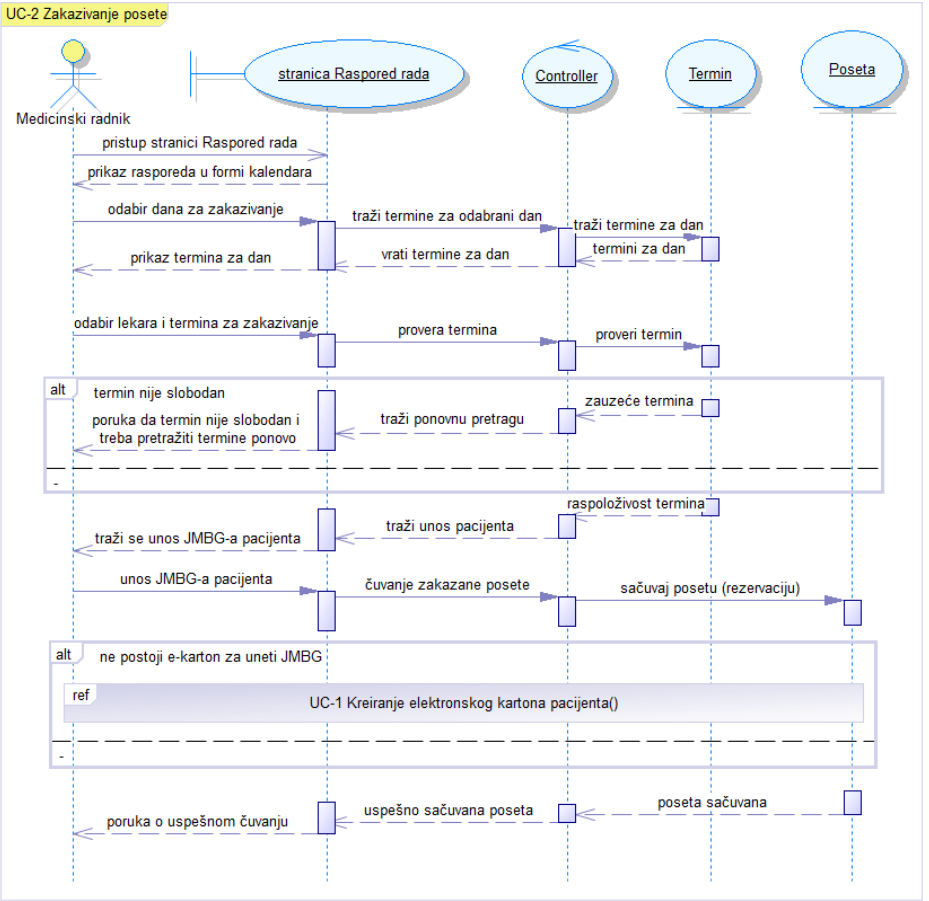


Slika 3.2 Dijagram aktivnosti za slučaj korišćenja UC-2 Zakazivanje posete [Izvor: Marina Damnjanović]

SEKVENCIJALNI DIJAGRAM

Elementi sekvencijalnog dijagrama: akter, objekat, linija života i poruka.

Sekvencijalni dijagrami služe za prikaz interakcije koja se dešava između različitih objekata u okviru jednog slučaja korišćenja. Oni predstavljaju objekte koji učestvuju u slučaju korišćenja i poruke koje se razmenjuju između njih u toku izvršenja tog slučaja korišćenja. Sekvencijalni dijagram je dinamički model koji prikazuje eksplicitnu sekvencu poruka koji se razmenjuju između objekata u definisanoj interakciji.
Elementi sekvencijalnog dijagrama: akter, objekat, linija života i poruka. Poruka može biti prosta i povratna.



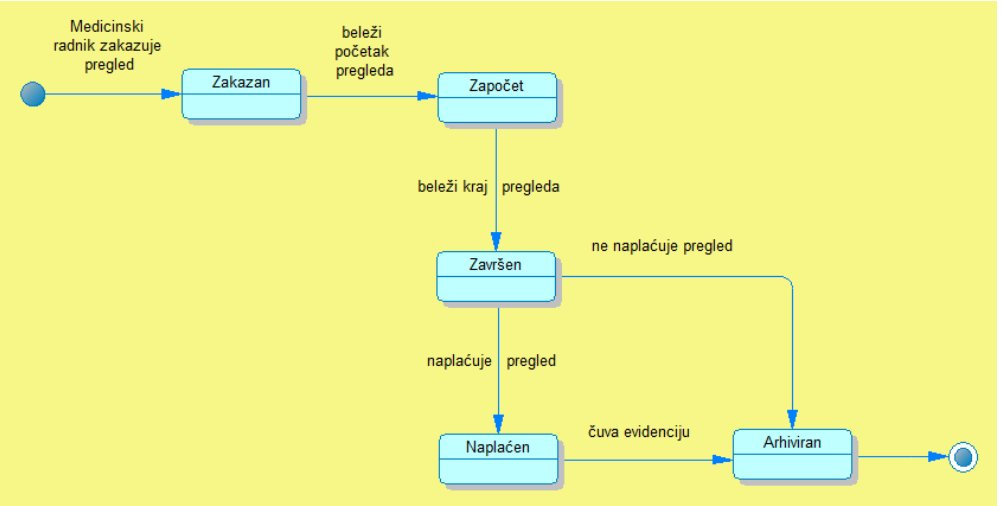
Slika 3.3 Sekvencijalni dijagram za slučaj korišćenja UC-2 Zakazivanje posete [Izvor: Marina Damnjanović]

DIJAGRAM STANJA

Elementi dijagrama stanja: stanje, događaji za prelazak stanja, početno stanje, krajnje stanje.

Stanje se može definisati kao skup vrednosti koje opisuju jedan objekat u specifičnom trenutku vremena i predstavlja tačku u životu objekta u kojem on mora da zadovolji izvesne uslove, izvrši neke akcije ili čeka na neke događaje. Stanje se menja kada objekat primi neki događaj; za objekat se tada kaže da je pretrpeo promenu stanja.

Dijagram stanja prikazuje kako objekti iz inicijalnog stanja (prikazuje se malim popunjenim krugom), pri pojavi izvesnih događaja ili zadovoljenjem izvesnih uslova, prelaze u druga stanja.
Elementi dijagrama stanja: stanje, događaji za prelazak stanja, početno stanje, krajnje stanje.



Slika 3.4 Dijagram stanja jednog pregleda u sistemu privatne klinike [Izvor: Marina Damnjanović]

ZADACI ZA VEŽBU

Tekst zadataka za vežbu

ZADATAK 1.

Po uzoru na primer sa slike 6 u poglavlju 2, nacrtati samostalno dijagram aktivnosti za slučaj korišćenja sistema privatne klinike *UC-3 Evidencija obavljene posete*, gde je primarni akter Medicinski radnik. (15 min)

ZADATAK 2.

Po uzoru na primer sa slike 11 u poglavlju 2, nacrtati samostalno stablo odlučivanja za slučaj korišćenja *UC-2 Zakazivanje posete lekaru*, kada pacijent traži da zakaže pregled u konkretnom terminu kod konkretnog lekara. Obratite pažnju na alternativne tokove i izuzetke koji su navedeni u opisu ovog slučaja korišćenja (pogledati vežbu br. 6) (15 min)

ZADATAK 3.

Nacrtajte sekvencijalni dijagram za oba slučaja korišćenja gde je akter lekar (pogledati urađenu vežbu br. 6) (20 min)

ZADATAK 4.

Na primeru onlajn prodavnice sportske opreme i suplemenata, uraditi sledeće:

- Nacrtati dijagram slučajeva korišćenja
- Nacrtati dijagram aktivnosti
- Nacrtati dijagram stanja za jedan proizvod
- Nacrtati sekvencijalni dijagram za slučaj korišćenja po izboru
- Nacrtati stablo odlučivanja za proces poručivanja jednog proizvoda

Poglavlje 4

Domaći zadatak

DOMAĆI ZADATAK 9

Tekst domaćeg zadatka

Za slučaj upotrebe koji ste napisali za DZ06, kreirajte sekvencijalni dijagram.

U skladu sa tim slučajem korišćenja i sistemom koji ste dobili za rad, odaberite još jedan od dijagrama koji su obrađeni u lekciji i nacrtajte odabrani dijagram.
Napomene:

Kao rešenje zadatka se šalju UML modeli (.oom fajlovi iz PowerDesigner alata).

Rešenje zadatka pošaljite na mejl adresu predmetnog asistenta. Rok za izradu je definisan Plan i programom predmeta.

Poglavlje 5

Projektni zadatak

ZADATAK ZA RAD NA PROJEKTU

Tekst zadatka za rad na projektu

Obratite pažnju na odeljak **Dodatak B: Modeli analize** u uzorku SRS dokumenta.

U svoj SRS dokument, koji ste kreirali nakon 8. nedelje nastave, uključite relevantne modele. Obavezno uključite sekvencijalne dijagrame (1 do 2, zavisno od veličine slučaja korišćenja za koji ga crtate). Sekvencijalni dijagram treba da sadrži glavni i alternativni tok, kao i izuzetke. Ako ima mogućnosti, uključite još neki model od ponuđenih: dijagram stanja, dijagram aktivnosti, stablo odlučivanja ili dijagram toka podataka.

Poglavlje 6

Zaključak

ZAKLJUČAK

1. Nije moguće jednim dijagramom opisati sve zahteve sistema, kao i softverske zahteve. Koriste se različiti dijagrami da bi se istakle posebna svojstva sistema
2. Analitičar dok sluša kupca uočiće ključne reči koje može da prevede u specifične elemente modela.
3. Često razvojni tim mora da koristi sve tipove modela analize. Najčešće izvrši izbor onih koji su odgovarajući za modeliranje najsloženijih i najrazličitijih delova sistema, kao i onih delova u kojima se najčešće javljaju dvosmislenosti i neizvesnosti.
4. **Dijagram konteksta** utvrđuje tzv. spoljne entitete van granica sistema koji su interfejsa sistemom
5. **Dijagram toka podataka** pokazuje kretanje podataka kroz sistem i procese njihove transformacije.
6. U inženjerstvu zahteva, koriste se za grafički opis scenarija prikazanih u slučajevima korišćenja. **UML sekvencijalni dijagrami** su posebno pogodni za ove namene.
7. **Trakasti dijagrami** pokazuju poslovni proces ili operacije predloženog poslovnog procesa. UML dijagram aktivnosti je sličan trakastom dijagramu, jer opisuje poslovni proces.
8. **UML dijagram stanja** predstavlja koncizan, kompletan, i nedvosmislen opis stanja nekog objekta u sistemu, kao i uslova za promenu tih stanja.
9. **Tabela stanja** prikazuje sve moguće tranzicije između stanja u vidu matrice.
10. **Mapa dijaloga** pokazuje elemente dijaloga i navigacione veze između njih.
11. **Tabela odlučivanja** opisuje šta sistem treba da radi i koje odluke da donese u određenim složenim uslovima.
12. **Stablo odlučivanja** dokumentuje grafičkim putem poslovna pravila koje treba ugraditi u softver.
13. **Tabela događaja i odziva** sistema navodi sve događaje i očekivano ponašanje sistema na takve događaje. Tabele događaja i odziva se najviše koriste kod sistema u realnom vremenu.
14. Razlika tradicionalnih i agilnih metoda razvoja pri modeliranju zahteva se odnosi samo na to kada se kreira neki model i nivo detalja koji on sadrži.

REFERENCE

Nastavni materijal pripremljen za studente se pravi s namerom da im omogući brži i skraćeni uvid u program lekcije, a na bazi jedne ili više referentnih udžbenika i drugih izvora . Nastavni materijal nije zamena za ove udžbenike, koje treba koristiti ako student želi da se

detaljnije upozna sa nastavnom materijom. Očekuje se od studenta da poseduje bar jedan od navedenih udžbenika u Planu i programu predmeta.

Ova lekcija je urađena na bazi teksta datom u **poglavlju 12** knjige: **Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013**. Za detaljnije proučavanje i primere, studentima se preporučuje da pročitaju ovo poglavlje. Manji uticaj na sadržaj lekcije imaju ostale reference navedene u Planu i programu predmeta,