



# Interakcija čovek računar (Human-Computer Interaction - HCI)

Elektronski fakultet Niš

Softverski alati za  
izradu korisničkog  
interfejsa



Elektronski fakultet u Nišu



## Sadržaj

- Uvod
- Metode za izradu specifikacije
  - Gramatike
  - Stabla menija
  - Dijagrami prelaza
  - Dijagrami stanja
- Softverski alati za izradu KI
  - Alati za skiciranje KI
  - Alati za razvoj softvera
- Softverski alati za evaluaciju KI
  - Alati za logovanje u realnom vremenu
  - Alati za automatsku analizu



Elektronski fakultet u Nišu



## Uvod

- Koriste se da bi se automatizovao razvoj KI
- Po prirodi su grafički – obezbeđuju brz razvoj interfejsa
- Koriste princip WYSIWYG
  - *What you see is what you get*
  - Korisnik se oslobađa potrebe da zamišlja kako će da izgleda rezultat
- Koriste princip direktne manipulacije
  - Palete osnovnih komponenti sa kojih se prevlače odgovarajuće komponente na radnu površinu
  - Direktna promena svojstava na radnoj površini ili preko liste osobina
  - Funkcionalnost se dodaje kreiranjem metoda koji obrađuju događaje koje generišu komponente



Elektronski fakultet u Nišu



## Metode za izradu specifikacije

- Prvi korak u razvoju KI je njegovo specificiranje
- Podrazumevani jezik za pisanje specifikacija je prirodni jezik i skice
  - Duge, nejasne i dvosmislene specifikacije
  - Treba da budu korektne, konzistentne i kompletne
- Alternativa prirodnom jeziku su formalni jezici za koje se definišu gramatike
  - Automatska provera validnosti iskaza
  - Kod KI se koriste za opisivanje sekvenci akcija

## Gramatike

- Moćno sredstvo za specifikaciju tekstualnih komandi ili izraza koje neki program treba da razume
- Nezamenjive kod interfejsa zasnovanih na komandnoj liniji
- Kod grafičkih KI ograničena primena na obradu simboličkih izraza (*spreadsheets*) i validaciju unosa
- Za definisanje formalnih jezika se koristi Bekusova normalna forma (BNF)

## Bekusova normalna forma (BNF)

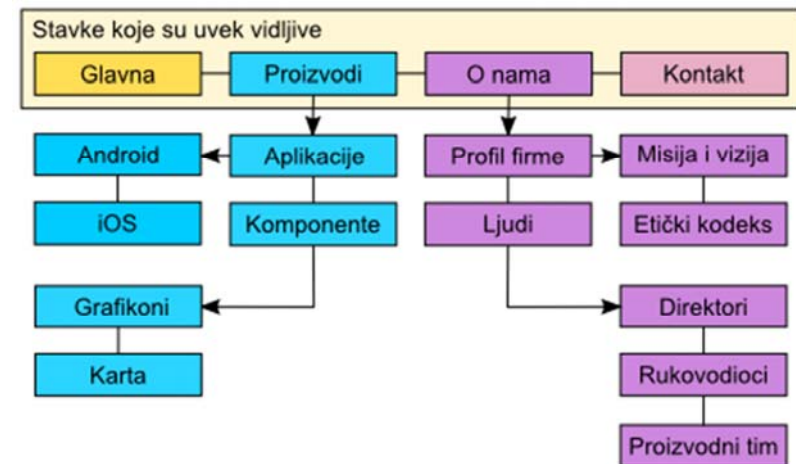
- Meta simboli BNF-a su:
  - ::= sa značenjem „je definisano kao“
  - | sa značenjem „ili“ i
  - <> koje se koriste da uokvire neterminale
- Primer definicije kontakta u telefonskom imeniku:

```
<Kontakt> ::= <Naziv> <Telefon>
<Naziv> ::= <Prezime> „,“ <Ime>
<Prezime> ::= <tekst>
<Ime> ::= <tekst>
<tekst> ::= <slovo> | <slovo><tekst>
<slovo> ::= "A" | "a" | "B" | "b" | ... | "Z" | "z"
<Telefon> ::= <kod_države> <kod_mesta> <lokalni_broj>
<kod_države> ::= "+" <cifra> <cifra> <cifra>
<kod_mesta> ::= <cifra> <cifra>
<lokalni_broj> ::= <cifra> <cifra> ... <cifra>
<cifra> ::= "0" | "1" | "2" | ... | "9"
```

## Stabla menija

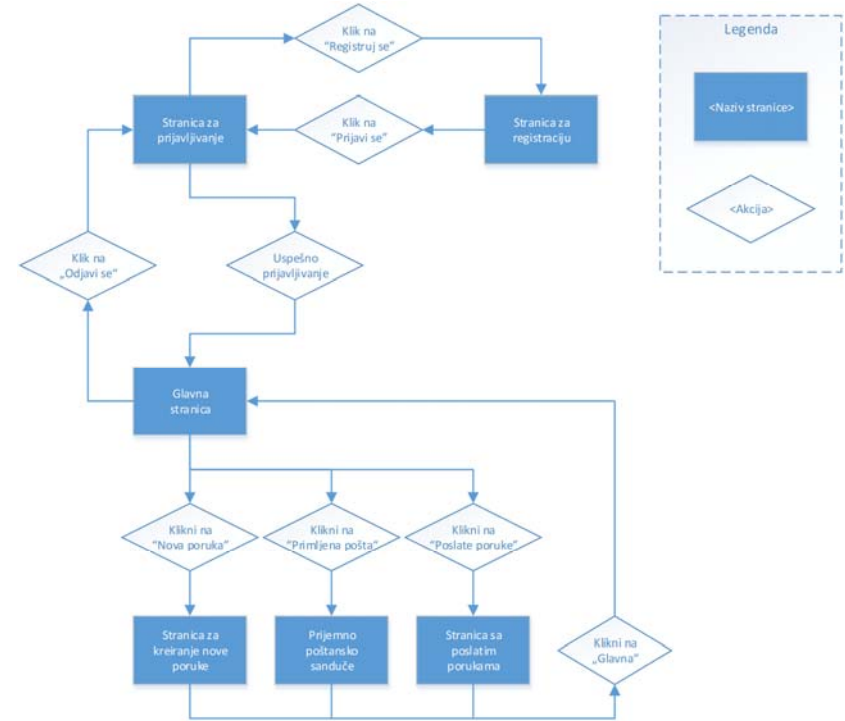
- Jednostavno sredstvo za pregled mogućnosti sistema
- Pruža uvid u kompletnu strukturu
  - Koristi se za proveru konzistentnost, kompletnosti, dvosmislenosti i redundanse
- Slično stablu menija moguće je formirati i stablo dijaloga

## Primer stabla menija



## Dijagrami prelaza

- Problem sa stablima menija je to što ne prikazuju celokupnu strukturu mogućih korisničkih akcija
  - Npr. nije moguće predstaviti da se neka stavka u meniju koristi da vrati korisnika na prethodni podmeni
  - Za neke aspekte razvoja je potrebno specificirati sve moguće prelaze
- Dijagrami prelaza predstavljaju graf
  - Čvorovi predstavljaju stanja
  - Potezi su moguće prelaze između ovih stanja
  - Svaki prelaz je definisan korisničkom akcijom koja ga inicira, a opciono i odgovorom sistema

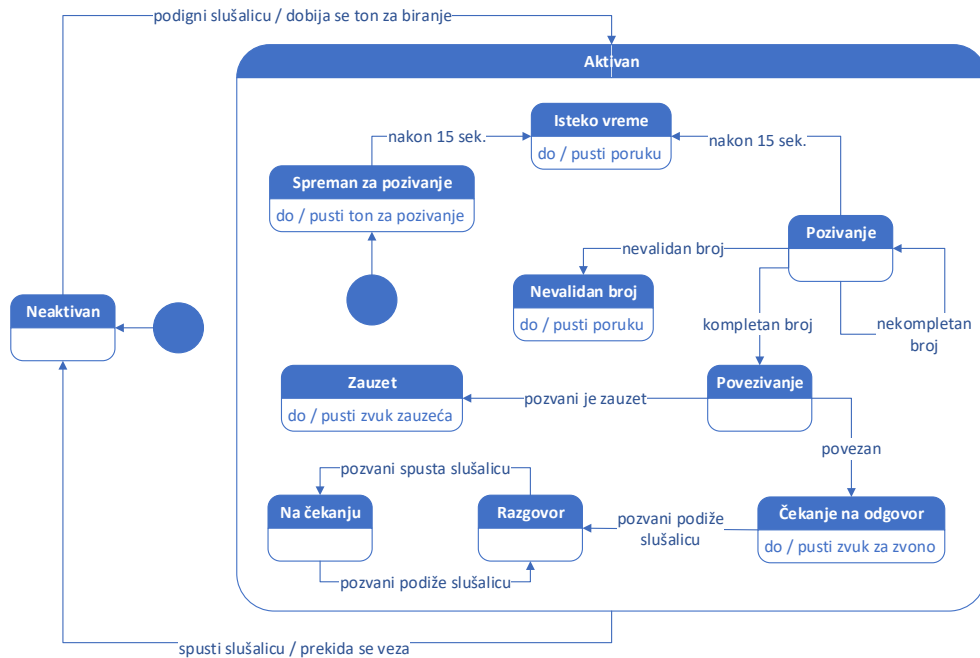


## Dijagrami stanja

- Mane dijagrama prelaza su što brzo narastaju, pa postaju zbunjujući
- Na složenost dijagrama prelaza takođe negativno utiču
  - veze od svih stanja sa stanjem gde se prikazuje pomoć
  - povratne veze između stanja kao rezultat akcije za vraćanje u nazad
  - veze svih stanja sa stanjem za gašenje aplikacije
- Imaju problem sa predstavljanjem konkurentnosti i sinhronizacije
- Alternativa koja rešava iskazane probleme su dijagrami stanja (statechart)

## Dijagrami stanja

- Dijagrami stanja nude mogućnost grupisanja podgrafova u podstanja
  - Prelazi koji se ponavljaju se mogu izvući van podgrafa i primeniti na podstanje
- Sadrže mehanizme za predstavljanje konkurentnih radnji i tačaka sinhronizacije
- Dijagrami stanja su deo UML specifikacije
  - Dostupni su kroz razne alate koji podržavaju UML
- Na narednom slajdu je dat primer UML dijagrama stanja za prikaz upotrebe telefona

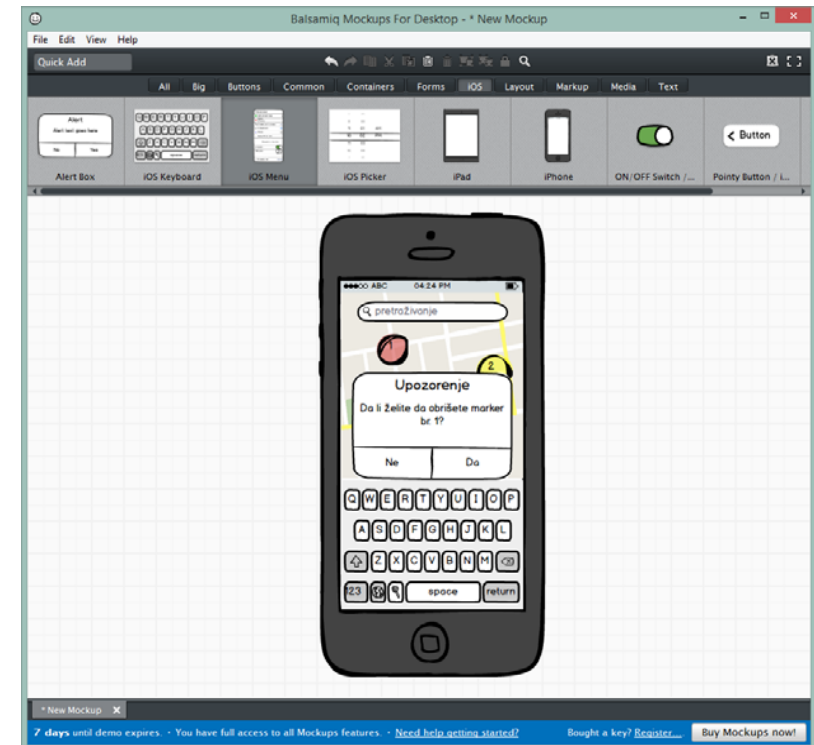


## Softverski alati iz izradu KI

- Koriste se za protipovanje i razvoj finalnog interfejsa
- Velika prisutnost alata ovog tipa na tržištu
- Vizuelno editovanje, odmah se vidi kakav će biti konačni rezultat
- Omogućavaju da domenski eksperti postanu projektanti KI
- Obezbeđuju odvajanje KI od programske logike
  - Brz razvoj KI i kasnije ažuriranje interfejsa

## Alati za skiciranje interfejsa

- U engleskom jeziku se koriste sintagme *wireframing tools* i *mockup tools*
- Skice su bitne u ranim fazama razvoja KI kako bi se ispitale alternative
- Mogu se raditi pomoću
  - papira i olovke
  - programa opšte namene (Word, PowerPoint, CorelDraw)
  - specijalizovanih alata (Balsamiq Mockups)
    - slika na narednom slajdu

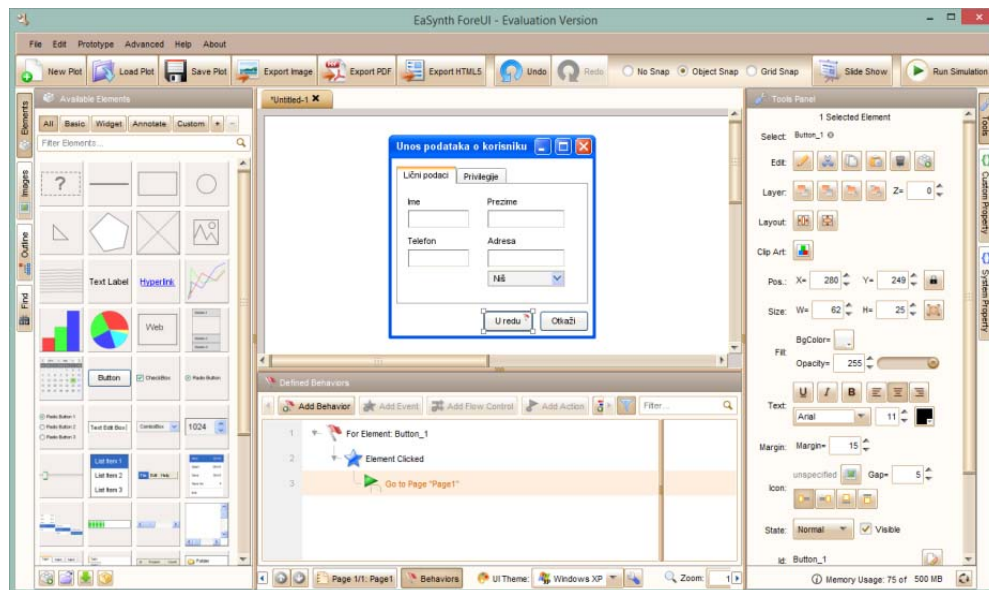


## Alati za skiciranje interfejsa

- Najjednostavnija primena alata za izradu skica je kreiranje niza statičkih slika (slajdova)
- Većina alata obezbeđuje i naprednije funkcionalnosti kao što su otvaranje i izbor stavki iz menija, klik na dugme, skrolovanje listi, prevlačenje ikonica i sl.
- Stvaraju realističniju sliku kako će interfejs za izgleda
- Skice korisničkog interfejsa prerastaju u prototipe
  - Nema sve funkcije aplikacije
  - Ne mora da uključi ni sve elemente interfejsa
  - Bitno je da realističnu prezentaciju kakav će interfejs da bude
  - Bitno da alat obezbedi mogućnost uvezivanja glavnih elemenata interfejsa
    - npr. da se klikom na neko dugme otvori neki dijalog

## ForeUI alat za prototipovanje KI

- Kreiranje i vizuelizacija korisničkih interfejsa u vidu skica, žičanog modela ili realnog izgleda
- Kreiranje stranica sa elementima korisničkog interfejsa i mogućnost njihovog pregledavanja u vidu sekvence slajdova
- Definisanje ponašanja za elemente interfejsa (npr. klik na dugme otvara neku stranicu) i generisanje simulacije interfejsa koja se izvršava u veb pregledaču
- Mogućnost proširivanja skupa ponuđenih elemenata kreiranjem sopstvenih elemenata ili uvozom dodatnih kolekcija



## Alati za razvoj softvera (1)

- **Nivo 4:** Alati visokog nivoa zasnovani na modelu
  - Obezbeđuju brz razvoj aplikacija određenog tipa
  - Primeri: *MS Access, Unity*
- **Nivo 3:** Alati za brz razvoj aplikacija
  - Obezbeđuju brz razvoj aplikacija korišćenjem namenskog grafičkog okruženja
  - KI se pravi prevlačenjem gotovih komponenti sa palete i kreiranjem metoda za obradu događaja
  - Primeri: *MS Visual Studio, Qt Creator, Eclipse i NetBeans IDE.*



## Alati za razvoj softvera (2)

- **Nivo 2:** Okviri za razvoj aplikacija
  - Softverske arhitekture, tj. biblioteke klasa, specijalno napravljene da podrže razvoj grafičkih KI
  - Sadrže klase za različite elemente korisničkog interfejsa
  - Primeri: Qt okvir, .NET Framework i MFC
- **Nivo 1:** Aplikacioni programski interfejs (API) operativnog sistema
  - Skup funkcija niskog nivoa pomoću kojih je moguće kreirati različite interfejsne elemente koje operativni sistem nudi ili vršiti iscrtavanja po ekranu
  - Primeri: Win32/GDI+, X11 Windowing System i Apple Quartz

## Kriterijumi za izbor alata (1)

- Uzimajući u obzir navedeni spisak slojeva, kao logična, nameće se preporuka da treba koristi alate što višeg nivoa.
- Međutim, sa povećanjem nivoa rastu i ograničenja koje alati nameću.
  - Npr. korišćenjem alata 4. nivoa ćete relativno brzo napraviti stereotipsku aplikaciju, ali raznovrsnost i mogućnost prilagođenja ovakvih rešenja nije velika
- Upotreba pravog alata predstavlja kompromis između sledećih 6 glavnih kriterijuma:
  - **NAREDNA 2 SLAJDA >>>**

## Kriterijumi za izbor alata (2)

- **Koji deo aplikacije se pravi**
  - Neki alati dopuštaju samo pravljenje prezentacionog dela aplikacije, dok drugi obezbeđuju mnogo širu podršku za celokupnu funkcionalnost
- **Vreme učenja**
  - Vreme potrebno da se nauči neki alat može da varira
- **Vreme izrade**
  - Vreme potrebno da se napravi korisnički interfejs takođe varira od alata do alata

## Kriterijumi za izbor alata (3)

- **Nametnuta metodologija**
  - Postoje alati koji nameću metodologiju za pravljenje aplikacije, dok ima i onih koji su mnogo fleksibilniji u ovom pogledu
- **Komunikacija sa drugim podsistemima**
  - Kada alat u sebi integriše podršku za različite funkcionalnosti, to može značajno da pojednostavi razvoj aplikacije
- **Proširivost i modularnost.**
  - Aplikacije evoluiraju, a i nove aplikacije često zahtevaju korišćenje ranije razvijenih celina
  - Alati 4. nivoa i okviri za razvoj aplikacija najčešće promovišu dobru organizaciju softvera, dok rešenjima koja nastaju na osnovu ostale dve grupe alata često manjka proširivost i modularnost

## Uporedni pregled 6 kriterijuma za izbor alata i njihovih nivoa

Kriterijum / Nivo	Deo aplikacije	Vreme učenja	Vreme izrade	Nametnuta metodologija	Komunikacija sa drugim podsistemima	Proširivost i modularnost
4	Za specifični domen	Dugo	Kratko	Prvo specifikacija, pa vizuelni deo	Vrlo dobra za specifični domen alata	Vrlo dobra
3	Prezentacioni sloj, interakcija	Kratko (dani)	Kratko	Prvo vizuelni deo	Zavisi od alata	Loša
2	Prezentacioni sloj	Dugo (nedelje)	Dugo	Nema	Dobra	Dobra
1	Sve	Vrlo dugo (meseci)	Vrlo dugo	Nema	Vrlo dobra	Vrlo loša

## Softverski alati za evaluaciju KI

- Pored alata za izradu, prirodno je da se softverski alati koriste i za evaluaciju KI
- Nekada i vrlo jednostavni alati mogu biti vrlo korisni
- Provera pravopisa
- Jednostavne metrike koliko različitih ekrana, formi ili veza između njih imamo nam govore o veličini KI
- Složenije metrike se mogu koristiti za određivanje
  - da li je stablo menija previše duboko i da li poseduje redundansu,
  - da li su sve forme konzistentno naslovljene,
  - da li svi dugmići imaju pridodate prelaze, itd.

## Softver za logovanje u realnom vremenu

- Beleži šablone korisničke aktivnosti
- Izlaz ovog softvera su izveštaji o
  - učestanosti određenih poruka o greški,
  - korišćenju pojedinih stavki iz menija,
  - prikazivanja pojedinih dijaloga,
  - pozivanju pomoći i dr.
- Ove informacije su od ogromnog značaja za rafinisanje inicijalnog dizajna

## Softverski alati za automatsku analizu (1)

- Mogu se lako primeniti već u ranim fazama procesa razvoja
- Prosta merenja koja nam daju informaciju o boji kontrola po dijalogu, balansu između gornjeg i donjeg ili levog i desnog dela i sl.
- Korisna u pogledu stilskih informacija, ali su vrlo limitirana po pitanju detekcije anomalija
- Izveštaji o veličini margina i spisak korišćenih boja i fontova, iako prosti, mogu da nam detektuju nekonzistentnosti u dizajnu korisničkog interfejsa



## Softverski alati za automatsku analizu (2)

- Automatski alati za analizu su najveću primenu našli kod veb sajtova
  - Standardizacija jezika i formata koji se koriste na vebu doprinela tome
- Kod desktop razvojnih okruženja imamo mnoštvo jezika i okvira
  - Uslovio relativno odsustvo automatskih alata uprkos velikom broju metrika i pravila koji se javljaju u literaturi