

Interakcija čovek-računar

Računske vežbe – I čas

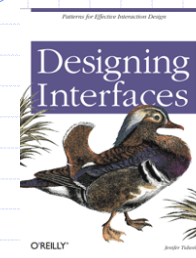
Uvod

- ◆ Nekada je razvoj interfejsa bio prilično usmeren, obzirom na ograničeni skup komponenti i pravila za njihovu primenu
- ◆ Danas je paleta komponenti i ideja neuporedivo veća, a korisnici naviknuti na interfejse konstruisane od strane timova profesionalnih dizajnera
- ◆ Da bi smo olakšali posao projektovanja interfejsa oslonićemo se na obrasce

Obrasce

- ◆ Obrasce su strukturne karakteristike i karakteristike ponašanja koje unapređuju prihvatljivost nečega, bilo da je to korisnički interfejs, Web sajt, objektno-orijentisani program ili zgrada.
- ◆ Obrasce povećavaju razumljivost i upotrebljivost
- ◆ Obrazac je opis najbolje prakse u okviru odgovarajućeg domena

Preporučena literatura



Designing Interfaces

Patterns for Effective Interaction Design

By Jenifer Tidwell

First Edition November 2005

Pages: 352

ISBN 10: 0-596-00803-1

ISBN 13: 9780596008031

Korisnici

- ◆ Dobar interfejs ne počinje slikama
- ◆ Počinje razumevanjem korisnika
 - **kakvi** su
 - **zašto** koriste softver
 - **kako** koriste softver
- ◆ Scenario komunikacije korisnika i računara i definisanje termina

Ispitivanje korisnika

- ◆ Prvi korak u projektovanju interfejsa je otkrivanje šta zapravo korisnik zeli da ostvari datom aplikacijom
- ◆ Postavljanje pravih pitanja pomaže u postizanju tog cilja
- ◆ Bez obzita na jasno postavljene zahteve, neophodno je ispitivanje korisnika. Po tome se razlikuju prosečne od zaista kvalitetnih aplikacija
- ◆ "Zašto" pristup u otkrivanju tipičnog scenarija

Upoznavanje korisnika

- ◆ Svaki korisnik je drugačiji
- ◆ Čak i isti korisnik u različitim situacijama očekuje različite stvari (može biti tolerantan na komplikovan interfejs u nekom specijalizovanom programu, ali neće prihvatiti telefon sa komplikovanim komandama)
- ◆ Kako upoznati "prosečnog" korisnika naše aplikacije?

Upoznavanje korisnika

- ◆ Direktno posmatranje (razgovor)
- ◆ Proučavanje rada (case study)
- ◆ Pregledi (dobijaju se značajni statistički podaci, ali nema direktnog kontakta)
- ◆ Personalizacija (nije metod prikupljanja podataka, već šta raditi sa njima kada smo ih skupili)- za svaku veliku grupu korisnika kreira se virtuelna osoba koja oslikava najvažnije aspekte te grupe

Motivacija korisnika da uči

- ◆ Rečnik aplikacije mora da odgovara profilu korisnika (obezbediti način da nauče najbitnije termine, ali ne očekivati preveliki entuzijazam pri učenju)
- ◆ Lako je preterati u proceni napora koji će korisnici biti spremni da ulože da bi savladali dati interfejs
- ◆ Za eksperte i napredne korisnike treba obezbediti interfejs koji omogućuje efikasnu korišćenje i dostupnost svih opcija u svakom trenutku (kriva učenja je vrlo strma i potrebno je i više meseci za savladavanje)
- ◆ Za povremene (*occasional*) korisnike interfejs mora biti maksimalno pojednostavljen uz čestu primenu wizarđa (vođenja korak-po-korak)
- ◆ Većina aplikacija nalazi se negde na pola puta i treba da obezbedi i jedno i drugo

Obrasci

1. Bezbedno istraživanje
2. Trenutno zadovoljenje
3. Dovoljnost
4. Promene u sredini toka
5. Odloženi izbor
6. Inkrementalna konstrukcija
7. Uobičajenost
8. Prostorna memorija
9. Potsetnik
10. Automatizovano ponavljanje
11. Samo tastatura
12. Saveti drugih ljudi

1. Bezbedno istraživanje

Dozvoli mi da se "šetam" po aplikaciji, bez da se izgubim ili upadnem u nevolje.

Primeri:

- ◆ Multi-level undo (primena transformacija i poništavanje npr. u Photoshopu)
- ◆ Back dugme (vraćanje na prethodnu stranu u web-browseru)

2. Trenutno zadovoljenje

Hoću da zadatak uradim sada, ne kasnije.

- ◆ Ljudi očekuju trenutni rezultat
- ◆ Ako nema pozitivnog iskustva (rezultata) posle par sekundi, verovatno odustaju od daljeg korišćenja

2. Trenutno zadovoljenje

- ◆ Ukoliko se može predvideti šta će novi korisnik prvo pokušati da uradi, predvideti interfejsom da to bude ekstremno olakšano
 - Ako korisnik želi nešto da napravi – početni ekran treba da sadrži praznu radnu površinu i toolbar sa potrebnim alatima
 - Ako korisnik treba da obavi neki zadatak – ukazati na početnu tačku i put kojim treba da ide
- ◆ Minimizovati sve što odvlači korisnika od "brzog starta" (registracija, dugi setovi instrukcija, slike koje se sporo učitavaju, reklame itd.)

3. Dovoljnost

To je dovoljno dobro. Ne želim da provedem više vremena u učenju da bi rezultat bio bolji.

- ◆ Vrlo retko se sistematski prilazi učenju interfejsa
- ◆ "Pretpostavljam da to dugme radi ono što mi treba"
- ◆ Korisnik obično brzo pregleda interfejs i bira ono što pro ugleda a misli da će ga dovesti do cilja

3. Dovoljnost

- ◆ Engleski termi *satisficing* je kombinacija *satisfying* (zadovoljavajući) i *sufficing* (biti dovoljan)
- ◆ Termin uvodi Herbert Simon 1957 objašnjavajući ponašanje ljudi u različitim ekonomskim i društvenim situacijama
- ◆ Ljudi uvek biraju "**dovoljno dobro**" umesto "**najbolje**", pogotovu ako to zahteva dodatno vreme i napor!

3. Dovoljnost - preporuke

- ◆ Sve tekstualne oznake treba da budu: kratke, jednostavne i lako čitljive (meniji, dugmići, linkovi i sl.)
- ◆ Omogućiti da prva korisnikova pretpostavka o značenju oznake bude tačna
- ◆ Ako više puta pogreši u proceni značenja – biće frustriran i vrlo verovatno će odustati od korišćenja programa

3. Dovoljnost - preporuke

- ◆ Omogućiti jednostavno i lako kretanje kroz interfejs i vraćanje na prethodno stanje
- ◆ Omogućiti "otvor za spašavanje" (kada se dovoljno "upetlja" da može doći na početno stanje)
- ◆ Komplikovani interfejs ima visoku spoznajnu cenu za nove korisnike. Korisnik obično traži prvu stvar koja mu se čini da bi mogla raditi.

4. Promene u srednjem toku

Predomislio sam se. Sada hoću da uradim ovo.

- ◆ U toku posla, korisnik može da se predomisli. Treba mu omogućiti promenu toka.

Primer: Prilikom pregledavanja sajta sa knjigama, može se uočiti neki link (reklama) koja nas može odvesti sasvim na drugu stranu.

4. Promene u srednjem toku

- ◆ Omogućiti izbor više opcija
- ◆ Korisnik ne sme biti "zaključan" bez mogućnosti izbora i/ili globalne navigacije, bez veze ka drugim stranicama ili funkcijama (osim ako nema posebnog razloga za to)
- ◆ Omogućiti prekidanje neke akcije, pamćenje stanja i ponovno vraćanje na tu tačku (*reentrance*)

4. Promene u srednjem toku

- ◆ Podrška za *reentrance* može se ostvariti kroz:
 - dijaloge koji pamte unete vrednosti,
 - modal prozore (mogu se odvući na stranu i ostaviti za kasniju upotrebu)
 - rad sa više dokumenata ili čak projekata (svi tekst-editori i alati za programiranje koriste ovaj model)

Vrlo su cenjeni sajtovi sa dugmetom "I'll finish it later", jer omogućuju korisniku da kasnije dopuni odgovarajuću formu.

5. Odložen izbor

Ne želim sada da odgovorim na to pitanje. Pusti me da završim posao.

- ◆ Ako se korisniku postavljaju nepotrebna pitanja dok pokušava da obavi neki posao, on će verovatno poželeti da odgovore ostavi za kasnije (direktno je u vezi sa obrascem 2 – Trenutno zadovoljenje)

Primer: Prijava preko Web sajta treba da omogući zadavanje samo neophodnih informacija (npr. ime i e-mail), a sve ostalo (strane i strane ličnih podataka) ostaviti za kasnije (ako korisnik ikada poželi da ih dopuni).

5. Odložen izbor - preporuke

- ◆ Ne treba u startu zatrpiti korisnika podešavanjima, pitanjima i izborima
- ◆ Obavezno naznačiti polja koja je neophodno uneti i smanjiti njihov broj na minimum (dopustiti nastavak bez popunjavanja opcionih polja)
- ◆ Odvojiti važna pitanja od manje važnih. Prikazati skraćenu listu. Sakriti dužu.

5. Odložen izbor - preporuke

- ◆ Postaviti podrazumevane vrednosti za sve što je moguće. Ali bez obzira što su unapred unete vrednosti, uzeti u obzir da korisnik ipak mora da ih pregleda u slučaju da želi da ih promeni.
- ◆ Omogućiti da se jednom postavljene vrednosti mogu kasnije promeniti ili vratiti na podrazumevane vrednosti
- ◆ Omogućiti korisnicima da pregledaju sajt pre nego što se registruju (primer: TurboTax 2005).

6. Inkrementalna konstrukcija

Dozvoli mi da promenim ovo. Ne, ne sviđa mi se. Dozvoli da ga promenim ponovo. E, to je već bolje!

- ◆ Ništa se ne stvara odjednom
- ◆ Čak i eksperti, polazeći od početka, metodično idu kroz proces modifikacije, sve dok ne budu zadovoljni rezultatom
- ◆ Proces konstrukcije se mogu realizovati: odozgo-naniže i odozdo-naviše, ali je, u svakom slučaju, bolje da se odvija inkrementalno u malim koracima, nego u nekoliko velikih

6. Inkrementalna konstrukcija

- ◆ *Builder* stil interfejsa mora da podržava stil posla koji treba da se obavi
- ◆ Omogućiti lako pravljenje malih komponenti jedne po jedne
- ◆ Interfejs treba brzo da reaguje na promene i snimanja
- ◆ Odziv interfejsa je vrlo bitan – stalno prikazivati korisnika kako izgleda i ponaša se “cela stvar”

6. Inkrementalna konstrukcija

- ◆ Ako se prave alatke za programiranje ili simulaciju, maksimalno skratiti vreme kompajliranja da bi korisnik što pre video kako program “radi” nakon uvedenih promena
- ◆ Ako korisnik mora da čeka makar i pola minuta da bi video efekat inkrementalne promene, to može da naruši njegovu koncentraciju i tok (Mihaly Csikszentmihalyi razvio je teoriju *toka* u koga aktivnost uvlači korisnika)

7. Uobičajenost

To tako svuda radi. Zašto nije tako i ovde?

- ◆ Kada se često koristi interfejs, neke reakcije postaju refleksne (npr. Ctrl-S za snimanje)
- ◆ Time se povećava efikasnost i stvara efekat *toka*
- ◆ Ako se promene standardne komande, korisnik ponovo mora da razmišlja o interfejsu i time narušava *tok* (a refleks može izazvati i neku destruktivnu pojavu, zavisno od akcije koju izvršava)

7. Uobičajenost

Standardne komande:

- ◆ Ctrl-A – Select All
- ◆ Ctrl-C – Copy
- ◆ Ctrl-V – Paste
- ◆ Ctrl-X – Cut
- ◆ Ctrl-S – Save
- ◆ Ctrl-P – Print

7. Uobičajenost

- ◆ Promena standardnih opcija može da izazove štetu i ukoliko postoje upozoravajuće poruke (*confirmation dialog*), jer ih korisnik može refleksno zatvoriti pritiskom Enter tastera. Pogotovu ako se često javljaju, osim što postaju iritantne, korisnik može da previdi šta piše u njima
- ◆ Neke aplikacije pseudoslučajno postavljaju dugmiće i/ili tekst u upozoravajućim porukama za iste akcije, kako bi korisnik morao da pročita šta znači dugme na koje treba da klikne

8. Prostorna memorija

Kunem se da je dugme bilo ovde. Gde je nestalo?

- ◆ Kada ljudi manipulišu objektima i dokumentima, kasnije ih ne nalaze po njihovom imenu, već na osnovu toga gde su ih ostavili
- ◆ Mnogi koriste desktop za ostavljanje aplikacija i drugih stvari koje često koriste. Kada se ikonice nagomilaju prostorno pamćenje gde je šta pomaže da se brzo dođe do cilja
- ◆ Često se koristi termin "organizovani haos" za pojavu kada ljudi, bez obzira što sve izgleda nabacano, lako mog da pronađu ono što im je potrebno u masi drugih stvari

8. Prostorna memorija

- ◆ Dodavanje nove stavke u interfejsu nije problem, ali je problem reorganizovanje postojećih kontrola, obzirom na narušavanje prostorne memorije
- ◆ I zbog navika i zbog prostorne memorije poželjna je konzistentnost u interfejsu, čak i između različitih aplikacija, jer ljudi očekuju da slične stvari nađu na sličnim mestima

8. Prostorna memorija - preporuke

- ◆ Ako postoji mali broj objekata, dozvoliti korisniku da ih sam rasporedi, kao na desktopu (za veliki broj to ne važi)
- ◆ Ne uređivati raspored osim ako korisnik to eksplicitno ne traži
- ◆ Dinamički meniji mogu predstavljati problem, jer je korisnik navikao da nešto vidi na određenom mestu
- ◆ Početak i kraj liste i menija su "specijalne lokacije" koje se lako uočavaju i pamte (mnogo više nego sredina), pa ih pažljivo treba menjati (ili ih ne menjati uopšte)

9. Podsetnik

Staviću to ovde, da me podseti da to kasnije uradim.

- ◆ Obzirom da vodimo složene, visoko isplanirane, *multitask* živote, često je potreban podsetnik da bi obavili sve predviđene zadatke
- ◆ Postoji specijalizovani softver koji služi kao podsetnik, npr. Outlook, koji kalendarima, alarmima i na druge načine pamti, planira i signalizira odgovarajuće događaje

9. Podsetnik

Drugi načini:

- ◆ Poruke u obliku virtuelnih nalepnica
- ◆ Prozori ostavljeni na ekranu
- ◆ Zapisi direktno uneti u dokument (npr. "Završi me!")
- ◆ *Bookmars* u pretraživačima, za sajtove koje treba kasnije obići
- ◆ Dokumenti ostavljeni na desktopu umesto na uobičajenom mestu
- ◆ Pošta ostavljeni u mailbox-u
- ◆ ...

9. Podsetnik

- ◆ Prethodni načini za podsećanje ništa ne znače odgovarajućim aplikacijama (mejl-klijentu to je samo još jedan folder, tekst procesoru – samo još dopisanog tekst (Da li to što je tekst ogroman, ljubičast i na specijalnom mestu nešto znači Wordu? – Sigurno NE!))
- ◆ Projektanti softvera treba da podrže ovakve aktivnosti

9. Podsetnik - preporuke

- ◆ Sačuvati podatke u privremeno zatvorenim i napola završenim dokumentima da bi se omogućio dalji rad
- ◆ Zapamtiti listu nekoliko poslednjih dokumenata koji su editovani
- ◆ Dodati posebnu listu (bookmark-ova) objekata od interesa
- ◆ Prepoznati da je zadatak ostao nezavršen i sa sledećim otvaranjem programa ukazati na njega
- ◆ Objediniti različite izvore koji mogu služiti kao podsetnik (e-mail, dokumenti, kalendari itd.). Kreativnost nema granica!

10. Automatizovano ponavljanje

Koliko puta treba da ponovim ovo?

- ◆ Nekada se akcije moraju ponavljati više puta. Što je to lakše uraditi - to bolje!
- ◆ Idealno je da ponavljanje akcije svede samo na jedan klik
- ◆ Još bolje je da se serija ponavljanja može obaviti samo sa nekoliko klikova

10. Automatizovano ponavljanje

- ◆ Find-Replace dijalog je dobar primer automatizovanog ponavljanja (zamenjivanje jednog po jednog pojavljivanja, ili Replace All)
- ◆ Copy-Paste (smanjuje kucanje)
- ◆ Snimanje akcija (Photoshop, Corel, ...) i pravljenje makroa
- ◆ Skriptovana okruženja su još generalnije rešenje, ali korišćenje shell-ova i pisanje skriptova zahteva od korisnika prilično visok nivo programiranja
- ◆ Shortcut-ovi za često korišćene akcije (upotreba tastature) – Treba posmatrati korisnike da bi se uočile radnje koje se često ponavljaju kako bi se pružila podrška (potrebno je posmatranje, a ne razgovor, jer korisnici najčešće tih akcija nisu svesni).

11. Samo tastatura

Bez miša, molim!

- ◆ Neki korisnici imaju problema sa upotrebom miša, drugi smatraju da (ili zaista) brže rade ako ne posežu za mišem, dok treći čak i ne vide ekran, pa je tastatura jedini interfejs
- ◆ Zato je poželjno da se sve akcije mogu izvesti i samo preko tastature (ne isključujući upotrebu miša)

11. Samo tastatura

Standardne tehnike:

- ◆ Definisanje *shortcut*-ova preko tastature, akceleratora i mnemonika za pristup opcijama iz menija (npr. Ctrl-S)
- ◆ Selekcija iz liste (jednostruka i višestruka) korišćenjem strelica i kontrolnih tastera (Shift, Control i sl.)
- ◆ Tab taster za prebacivanje fokusa na sledeću kontrolu, Shift-Tab za kretanje unazad
- ◆ Većina kontrola (radio, combo,...) mogu menjati vrednosti korišćenjem strelica, Enter i Space tastera
- ◆ Većina dijaloga ima "podrazumevano" dugme (Submit, Done, OK i sl.) koje se aktivira samo pritiskom tastera Enter. Ono zatvara formu, prelazi na sledeću stranicu ili izvršava neku drugu podrazumevanu radnju

11. Samo tastatura

- ◆ Vrlo je teško upravljati grafičkim editorima samo preko tastature (mada ne i nemoguće)
- ◆ Korišćenje samo tastature preporučuje se samo za aplikacije koje su orijentisane unosu velike količine podataka (neke od takvih aplikacija ne zahtevaju čak ni Tab taster, već automatski prebacuju fokus na odgovarajuću kontrolu u skladu sa tokom unosa)

12. Saveti drugih ljudi

Šta bi drugi rekli o ovome?

- ◆ Društvena zajednica ima značaja i u projektovanju interfejsa
- ◆ Saveti mogu biti direktni i indirektni, mogu predstavljati komentare korisnika, ali i komentare i zahteve ključnih ljudi (naručioca posla, direktora, nadređenih i sl.)
- ◆ Sa druge strane, Internet pruža neiscrpni izvor primera (i gotovog koda) koji se mogu uključiti u vaš projekat (uvek treba razmotriti i ponuđena gotova rešenja)

Pa, krenimo u dizajn!

