



Mlađan Jovanović
Aleksandar Jevremović

INTERAKCIJA ČOVEK-RAČUNAR

UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I RAČUNARSTVO
TEHNIČKI FAKULTET

Mladen Jovanović
Aleksandar Jevremović

INTERAKCIJA ČOVEK - RAČUNAR

Prvo izdanje

Beograd, 2020.

INTERAKCIJA ČOVEK - RAČUNAR

Autori:

Doc. dr Mlađan Jovanović

Prof. dr Aleksandar Jevremović

Recenzenti:

Prof. dr Mladen Veinović

Prof. dr Milan Milosavljević

Izdavač:

UNIVERZITET SINGIDUNUM

Beograd, Danijelova 32

www.singidunum.ac.rs

Za izdavača:

Prof. dr Milovan Stanišić

Priprema za štampu:

Mlađan Jovanović

Dizajn korica:

Aleksandar Mihajlović, MA

Godina izdanja:

2020.

Tiraž:

300 primeraka

Štampa:

Caligraph, Beograd

ISBN: 978-86-7912-738-9

Copyright:

© 2020. Univerzitet Singidunum

Izdavač zadržava sva prava.

Reprodukcija pojedinih delova ili celine ove publikacije nije dozvoljena.

Predgovor

Udžbenik je primarno namenjen studentima 4. godine Tehničkog Fakulteta i Fakulteta za Informatiku i Računarstvo. Razvojem i unapređenjem tehnologije, tema interakcije čoveka i računara je aktuelna i sveprisutna u različitim oblastima informatike i naukama poput psihologije, medicine, jezika, sociologije i drugih. Pored toga, oblast prožima različite aspekte naših života, privatne i poslovne. Možemo reći da osim što mi kao ljudi oblikujemo tehnologiju, tehnologija zauzvrat oblikuje nas i naše živote. Zbog ove uzajamne relacije oblast ima značajno mesto u savremenom društvu.

Prethodno nas je inspirisalo u pisanju ovog udžbenika, kako bi materija bila razumljiva i prihvatljiva ne samo našim studentima, već i široj publici. U nadi da će im sadržaj biti koristan ne samo da kvalitetnije upotrebljavaju ili kreiraju tehnologiju, već i da bolje razumeju i sagledaju potrebe drugih i svoje.

Udžbenik najpre daje širi kontekst oblasti kroz njen razvoj i postulate. Zatim opisuje načine na koje koristimo tehnologiju u svakodnevnom životu. Nakon toga opisujemo principe korišćenja tehnologije sa aspekta cilja koji je zajednički svima nama, a to je da je to što koristimo lako i prijatno za upotrebu. Razumevanju principa doprinosi poznavanje osnovnih psiholoških mehanizama korišćenja tehnologije. Sledi opis razvoja interaktivnih tehnologija koji poštuje prethodno opisane principe. Najpre predstavljamo proces razvoja kao celinu, a zatim detaljnije obrađujemo njegove delove kao što su definisanje korisničkih zahteva, protopiziranje i testiranje interaktivnih tehnologija. Na kraju analiziramo korisničke i tehničke detalje grafičkih korisničkih interfejsa putem kojih najčešće upotrebljavamo tehnologiju.

Zahvaljujemo se našim studentima i recenzentima na savetima i sugestijama za unapređenje udžbenika.

Autori,
Beograd, Novembar 2020.

Sadržaj

1. Interakcija čovek-računar: uvod, istorijat i koncepti.....	5
1.1. Definicija.....	6
1.2. Razvoj.....	7
1.3. Ciljevi dizajna Interakcije Čovek-Računar	13
1.3.1. Ciljevi upotrebljivosti.....	15
1.3.2. Ciljevi doživljaja korisnika	17
1.4. Upotrebljivost i principi dizajna.....	19
1.4.1. Vidljivost.....	20
1.4.2. Povratna informacija	20
1.4.3. Ograničenja	21
1.4.4. Mapiranje, mentalni model i metafore	23
1.4.5. Konzistentnost.....	24
1.4.6. Priuštivost.....	25
2. Tipovi i stilovi interakcije	26
2.1. Tipovi interakcije	26
2.1.1. Instrukcije.....	27
2.1.2. Konverzacija.....	29
2.1.3. Manipulacija i navigacija	31
2.1.4. Eksploracija i pretraživanje	33
2.2. Stilovi interakcije	34
3. Korisnost i upotrebljivost.....	40
3.1. Lakoća učenja.....	40
3.1.1. Prepoznavanje i pamćenje	43
3.1.2. Konceptualni modeli	45
3.1.3. Smernice dizajna za lakoću učenja.....	46
3.2. Efikasnost	46

3.2.1. Prečice i podrazumevane vrednosti.....	48
3.2.2. Istorijat i agregacija.....	50
3.2.3. Smernice dizajna za efikasnost	52
3.3. Greške.....	53
3.3.1. Vrste grešaka	53
3.3.2. Prevencija grešaka.....	56
3.3.3. Poruke o grešci	57
3.3.4. Smernice dizajna za sigurnost	59
4. Osnove psihologije korisnika	60
4.1. Model procesiranja čoveka.....	60
4.2. Pažnja	62
4.3. Percepcija	65
4.4. Geštalt principi vizuelne percepcije	69
4.4.1. Princip blizine	70
4.4.2. Princip sličnosti	71
4.4.3. Princip kontinuiteta	74
4.4.4. Princip ograničavanja.....	75
4.4.5. Princip simetrije	76
4.4.6. Princip figura/pozadina	77
4.4.7. Princip pokreta	81
5. Dizajn orijentisan ka korisniku	83
5.1. Faze i karakteristike dizajna interaktivnog sistema.....	83
5.2. Model razvoja softvera.....	86
5.3. Spiralni model razvoja korisničkog interfejsa.....	87
5.4. PACT analiza	93
6. Definisane korisničkih zahteva	99
6.1. Vrste korisničkih zahteva	99
6.2. Prikupljanje podataka o korisnicima	101

6.2.1. Analiza dokumentacije ili postojećih rešenja	102
6.2.2. Posmatranje	103
6.2.3. Upitnik.....	103
6.2.4. Smernice za dizajn upitnika	106
6.2.5. Intervju i fokus grupa	109
6.2.6. Smernice za intervju.....	111
6.2.7. Logovanje podataka	112
6.2.8. Smernice za tehnike prikupljanja podataka o korisnicima.....	113
6.3. Obrada prikupljenih podataka	114
7. Prototipiziranje i dizajn	118
7.1. Prototipi - definicija i motivacija	118
7.2. Prototipi niske verodostojnosti.....	125
7.3. Prototipi visoke verodostojnosti.....	129
7.4. Ciljevi i analiza prototipa	132
8. Testiranje sa krajnjim korisnicima	135
8.1. Principi i vrste testiranja.....	135
8.1.1. Kontrolisani uslovi sa korisnicima	138
8.1.2. Prirodno okruženje sa korisnicima	140
8.1.3. Okruženje bez korisnika.....	141
8.2. Relevantni aspekti testiranja.....	142
8.2.1. Informisanje i pristanak korisnika.....	142
8.2.2. Faktori interpretacije rezultata	142
8.3. Testiranje upotrebljivosti.....	144
8.3.1. Elementi testiranja upotrebljivosti - tehnike, zadaci i korisnici	144
8.3.2. Elementi metodologije testiranja.....	157
8.4. Praćenje pogleda korisnika i kursora miša	162
8.5. Primena elektroencefalografije	168
8.6. Integracija podataka	169

8.7. A/B testiranje.....	172
9. Heuristička evaluacija	174
9.1. Heuristike upotrebljivosti	174
9.1.1. H1: Vidljivost statusa sistema	177
9.1.2. H2: Usaglašenost sistema sa realnim svetom.....	178
9.1.3. H3: Kontrola i sloboda u korišćenju	182
9.1.4. H4: Konzistentnost i standardi	183
9.1.5. H5: Prevencija grešaka	185
9.1.6. H6: Prepoznavanje pre pamćenja	185
9.1.7. H7: Fleksibilnost i efikasnost korišćenja	187
9.1.8. H8: Estetski i minimalistički dizajn	194
9.1.9. H9: Prepoznavanje i oporavak od grešaka	201
9.1.10. H10: Pomoć i dokumentacija	201
9.2. Elementi heurističke evaluacije.....	201
10. Vrste i elementi GKI	203
10.1. Grafički korisnički interfejs	204
10.1.1. Čovekovo vidno polje i grafički prikaz.....	205
10.1.2. Sistemska arhitektura grafičkog korisničkog interfejsa	206
10.2. Elementi GKI	214
10.2.1. Ulazni elementi	215
10.2.2. Navigacioni elementi.....	222
10.2.3. Informativni elementi	228
10.2.4. Kontejnerski elementi	230
10.3. Principi dizajna korisničkog interfejsa.....	235
11. Literatura	238
Spisak slika.....	244

1. Interakcija čovek-računar: uvod, istorijat i koncepti

Različiti interaktivni sistemi prisutni su u svakodnevnoj upotrebi. U toku jednog dana, prosečan korisnik susreće se sa mnoštvom uređaja i aplikacija koje predstavljaju sastavni deo svakodnevnice - od prenosivih uređaja, kao što su telefon, notebook računar, ili tablet, preko javnih uređaja, kao što su bankomati, mašine za kafu i aparati za zakazivanje, do uređaja i aplikacija kao što su štampači, kućni uređaji, računari u automobilima, sajtovi za kupovinu, rezervacije, učenje, umrežavanje, igre i drugi.

U opštem slučaju, lista uređaja i aplikacija koje koristimo je beskraja zbog toga što se kontinuirano susrećemo i sa novim tehnologijama i aplikacijama. Ono što je zajedničko svima njima i važno nama, kao korisnicima, jeste da su svi oni laki za upotrebu, odnosno korišćenje. Lako za korišćenje konkretno znači da jednostavno i brzo učimo da koristimo uređaj ili aplikaciju, da brzo koristimo kada naučimo, da pravimo malo grešaka i osećamo se sigurno prilikom korišćenja i, na kraju, da se osećamo prijatno u toku korišćenja.

Sa druge strane, čest je slučaj da imamo poteškoće prilikom interakcije sa mnogim proizvodima kako bi postigli naše stvarne ciljeve. Tada možemo reći da ovi proizvodi nisu dizajnirani tako da se mogu jednostavno koristiti u skladu sa našim navikama i očekivanjima. Tipično, ovakvi proizvodi su kreirani prvenstveno kao sistemi koji realizuju specifičan skup funkcija koje korisnici vide i koriste kao servise. Posledica ovakvog pristupa je da dok ovi sistemi implementiraju funkcije u skladu sa svojom namenom, njihovi korisnici često imaju problema sa njihovim korišćenjem.

Oblast Interakcija Čovek-Računar (IČR) ima za cilj rešavanje problema korišćenja interaktivnih sistema tako što definiše pravila, tehnike i metode koja se primenjuju u toku njihovog dizajna kako bi unapredili upotrebljivost. U širem smislu, pod pojmom interaktivni sistem podrazumevamo bilo koji uređaj, tehnologiju, alat, okruženje ili aplikaciju namenjene za direktno korišćenje od strane korisnika. Udžbenik se fokusira na grafički korisnički interfejs (GKI). U principu, oblast IČR izučava metodologiju i pravila razvoja interaktivnih sistema koji su laki, efikasni i prijatni za korišćenje.

U ovom poglavlju je dat osvrt na oblast IČR. Najpre je definisana oblast i njeni relevantni aspekti kroz hronološki razvoj i primere dizajna. Zatim su bliže određeni učesnici i faze dizajna interaktivnih proizvoda. Na kraju su opisani ključni elementi upotrebljivosti i njihovo korišćenje u dizajnu i evaluaciji interaktivnih tehnologija.

1.1. Definicija

Zvanična definicija oblasti IČR glasi:

“Disciplina koja se bavi dizajnom, evaluacijom i implementacijom interaktivnih računarskih sistema za ljudsku upotrebu i izučava glavne fenomene u vezi sa ovim sistemima.” (ACM SIGCHI 1996, p. 5) [1]

Kao što se vidi iz definicije, faze životnog ciklusa proizvoda - dizajn, implementacija i evaluacija - su razdvojene. Dizajn podrazumeva kreiranje prototipa budućeg sistema različitih nivoa veroostojnosti kako bi se u određenom trenutku, odnosno kada prototip kao takav zadovolji zahteve korisnika, krenulo sa implementacijom. Razlog je praktične prirode - faktor krajnjeg korisnika usložnjava sam proces razvoja i na ovaj način se smanjuju troškovi i dolazi do kvalitetnijeg krajnjeg proizvoda tako što se korisnici uključuju u proces dizajna. Termin dizajn se u nekim slučajevima koristi da označi i faze dizajna i implementacije. Sam razvoj je iterativne i inkrementalne prirode. Triplet faza se odvija u većem broju iteracija, gde se u svakoj iteraciji sistem obogaćuje izmenom postojećih ili dodavanjem novih funkcionalnosti.

Zvanični cilj oblasti je:

“Razvoj ili poboljšanje korisnosti, efektivnosti, efikasnosti i upotrebljivosti računarskih sistema” (Interacting with computers 1989, p. 3) [2]

Dizajn upotrebljivih interaktivnih proizvoda zahteva prethodno definisanje ciljne grupe korisnika, odnosno kome je proizvod namenjen i u kakvom kontekstu će biti korišćen. Kontekst podrazumeva vrstu uređaja (na primer mobilni telefon, tablet, stoni računar ili kombinovano), vrstu okruženja (na primer posao, prostor za stanovanje, zatvorena prostorija, otvoren prostor) i uslove u okruženju (na primer osvetljenje, nivo buke, prisustvo drugih ljudi). Sledeći važan aspekt predstavljaju vrste aktivnosti koje korisnici obavljaju dok interaguju sa proizvodom. Na primer, izbor i raspored specifičnih elemenata GKI zavisi od vrste aktivnosti koje podržava konkretna aplikacija. Na taj način, aplikacija za elektronsku kupovinu ima drugačiju strukturu i sadržaj korisničkog interfejsa od aplikacije za upravljanje pametnim kućnim uređajima. Slično tome, ako je ciljna aktivnost komunikacija na daljinu, sistem mora omogućavati jednostavno kreiranje poruka (pisanih ili govornih). Pored toga, interfejs mora omogućavati jednostavno upravljanje porukama (ažuriranje, brisanje, čuvanje i druge operacije).

Skup aktivnosti korisnika koje određeni sistem podržava definišu osobinu koja se naziva korisnost (eng. *utility*). Praksa korišćenja interaktivnih sistema je

pokazala da je korisnost samo jedan od preduslova za uspešno korišćenje. Pored toga, važno je i da se aktivnosti što je moguće lakše realizuju korišćenjem sistema. Ova osobina interaktivnog sistema je poznata pod nazivom upotrebljivost (eng. *usability*). Opšti cilj dizajna interaktivnih sistema je efikasno i efektivno obavljanje aktivnosti korisnika. Pod pojmom efektivno, podrazumevamo kvalitetno obavljanje aktivnosti u pogledu njihovih krajnjih rezultata. Efikasno podrazumeva optimalno korišćenje resursa kao što su vreme i kognitivno naprezanje. Proces dizajna interaktivnih tehnologija podrazumeva:

- dobro poznavanje načina na koji korisnici obavljaju aktivnosti koje treba podržati,
- razmatranje unapređenja u obavljanju aktivnosti u odnosu na tekući način obavljanja,
- postizanje kvalitetnog doživljaja korisnika u toku korišćenja,
- uvažavanje potreba i želja korisnika i njihovo aktivno učešće u procesu dizajna,
- primena verifikovanih tehnika dizajna orijentisanih ka korisnicima.

Cilj ovog udžbenika je izučavanje ovih aspekata kroz primere dizajna interaktivnih rešenja. Akcenat je na razumevanju i definisanju potreba krajnjih korisnika i dizajniranju korisnih, upotrebljivih i privlačnih interaktivnih aplikacija, sa akcentom na GKI.

1.2. Razvoj

U ranim danima računarstva, 50-tih godina dvadesetog veka, razvijani su hardverski sistemi namenjeni isključivo inženjerima. Računarski interfejs bio je hardverski, sastavljen od fizičkih panela i prekidača koji kontrolišu skup registara. Vanevar Buš, istraživač i profesor na Masačusetskom Institutu za Tehnologiju (eng. Massachusetts Institute of Technology - MIT) u novinskoj kolumni detaljno je opisao koncept računara budućnosti zasnovanog na metafori radnog stola [3]. Računar budućnosti, zvani Memex (Slika 1), predviđa dva monitora za pregled sadržaja u centralnom delu. Ručice ispod monitora služe za navigaciju kroz prikazani sadržaj. Na korisnikovoj levoj strani se nalazi skener. Na desnoj strani se nalazi komandna table mikrofona i zvučnika za glasovnu komunikaciju sa računarom i drugim korisnicima. Pored toga, svetlosna olovka služi za pisanje po centralnim monitorima. Prostor u fiokama rezervisan je za memoriju računara.



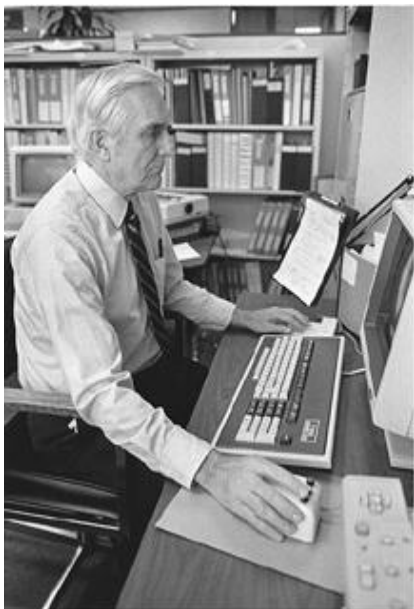
Slika 1. Memex - nacrt personalnog računara za obavljanje kancelarijskih poslova. Preuzeto iz [3].

Sedamdesetih i ranih osamdesetih godina prošlog veka pojavljuju se monitori, tada poznati kao vizuelne jedinice prikaza (eng. *visual display unit*, *VDU*) i personalni računari (eng. *personal computer*, *PC*). Tada računarski uređaji postaju dostupni i namenjeni širem skupu krajnjih korisnika i korisnički interfejs polako postaje kritičan faktor u korišćenju tih sistema. Nov koncept donosi niz izazova. Jedan od najvećih u to vreme bio je razvoj računara koji je pristupačan i upotrebljiv drugim korisnicima, a ne samo inženjerima. Kritični aspekt zajednički za različite aktivnosti prilikom korišćenja računara jeste angažovanje čovekovih kognitivnih mehanizama. Na primer, aktivnosti kao što su matematički proračuni, pisanje dokumenata, upravljanje korisničkim nalogima ili crtanje planova.

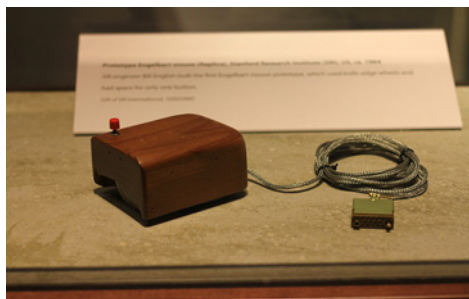
Da bi ovakve aktivnosti bile izvodljive korišćenjem računara, inženjeri računarstva i psiholozi uporedo su počeli da se bave različitim, ali komplementarnim elementima korisničkog interfejsa. Naučnici i programeri su se bavili tehnološkim elementima interfejsa tako što su razvijali programske jezike visokog nivoa za njihovo kreiranje (kao što su BASIC i Prolog), arhitekture sistema, metode dizajna softvera i komandne jezike za podršku navednim aktivnostima. Psiholozi su se bavili aspektima krajnjeg korisnika,

odnosno čoveka, tako što su pružali informacije o ljudskim sposobnostima kao što su percepcija, memorija, pažnja, donošenje odluke i druge.

Daglas Engelbart, istraživač sa Stanford Univerziteta, sa svojom grupom istraživača je 1968. godine izumeo višekorisnički softversko-hardverski sistem pod nazivom NLS (eng. *oNLine System*) [4]. NLS predstavlja suštinsku promenu u IČR. Pre svega što je preteča prvog operativnog sistema, a zatim i zbog toga što integriše uređaje za interakciju u jednu celinu (monitor, tastatura i miš). Sistem je omogućavao rad sa dokumentima pomoću skupa prozora, prvi je koristio hiperveze između dokumenata (eng. *hyperlink*), imao program za obradu teksta sa kontrolom verzija dokumenta, servis za lokalnu razmenu poruka nalik elektronskoj pošti, pa čak i aplikaciju za telekonferenciju sa deljenjem ekrana. Sve funkcionalnosti su bile dokumentovane elektronski (eng. *help*).



(a)

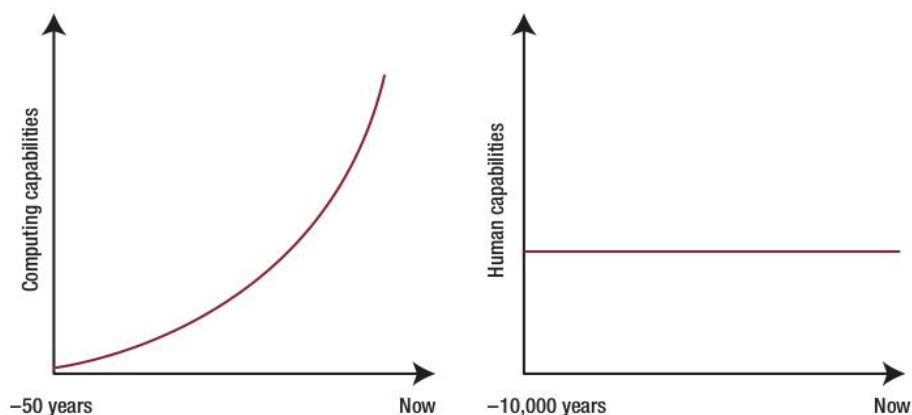


(b)

Slika 2. (a) Daglas Engelbart koristi NLS sistem; (b) uređaj miš koji je sistem koristio. Izvor: Michael Hicks from Saint Paul, MN, USA / CC BY.

Kontinuirano unapređenje mogućnosti interaktivnih računarskih tehnologija tog vremena, odnosno kombinacija vizuelnih displeja i interaktivnih tastatura imali su za posledicu da uređaji počinju da prevazilaze neke sposobnosti čoveka

(Slika 3). Na taj način su nastajali novi i usložnjavali se postojeći problemi korišćenja tehnologije. Istraživanje i razvoj GKI bilo je u povoju. Predmet istraživanja najpre su bile tehnologije za kancelarijsko poslovanje koje su se pojavile kao tehnologije za širu upotrebu. Istraživan je dizajn elemenata korisničkog interfejsa (eng. *widget*) kao što su meniji, prozori, palete i ikone, kako bi se ovi elementi što bolje rasporedili i prikazali u GKI.

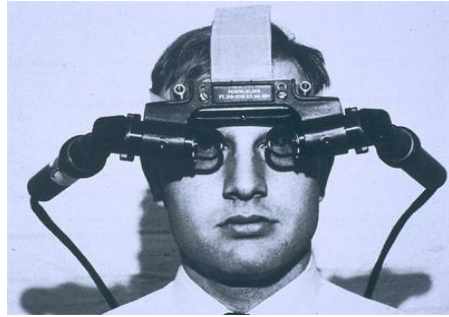


Slika 3. Uporedni pregled razvoja sposobnosti čoveka i računara. Preuzeto iz [5].

1963. godine Ivan Saterlend je u svojoj doktorskoj tezi predstavio Skečped (eng. *Sketchpad*) interaktivnu aplikaciju koja predstavlja prekretnicu u razvoju računarske grafike. Aplikacija je prva koja je korisnicima omogućavala modelovanje i crtanje u realnom vremenu, na displeju korišćenjem svetlosne olovke. Nakon toga, 1965. godine Saterlend i njegov student Bob Sproul se predstavili prvi sistem virtuelne stvarnosti, displej u obliku kacige koji se stavlja na glavu korisnika (eng. *Head-Mounted Display, HMD*).



(a)



(b)

Slika 4. (a) Demonstracija Skečped sistema; (b) Prva kaciga za virtuelnu stvarnost. Izvor: Dave Pape, CO, USA / CC BY.

Alan Kej se smatra se idejnim tvorcem modernog GKI zasnovanog na sistemu prozora sa unapređenim funkcijama kao što je preklapanje prozora. Njegov rad je implementiran u prvim komercijalnim personalnim računarima sa operativnim sistemima kao što su Xerox Star (1981) i Apple Macintosh (1984).



(a)



(b)

Slika 5. (a) Xerox Star radna stanica; (b) Apple Macintosh radna stanica. Izvor: Dave Pape, CO, USA / CC BY.

Sredinom 80-tih godina pojavljuju se nove tehnologije - prepoznavanje govora, multimediji, vizuelizacija informacija i virtualna stvarnost. Proširene mogućnosti ovih tehnologija donele su nove aplikacije za različite grupe krajnjih korisnika. Edukacija i trening su dva domena kojima je posvećena posebna pažnja. Interaktivna okruženja za učenje, edukativni softver i trening simulatori pojavljuju se kao nove aplikacije podržane ovim tehnologijama. Kreiranje novih interaktivnih sistema iziskivalo je drugačije veštine u poređenju sa onima koje su posedovali psiholozi i inženjeri. Na taj način i dodatne, specijalizovane profesije, kao što su edukativni tehnolozi, razvojni psiholozi i stručnjaci za obuku, ulaze u dizajn interaktivnih sistema.

Novi talas tehnologija se javlja 90-tih godina 20. veka: mrežni sistemi, mobilno računarstvo i senzorski uređaji. Step razvoja tehnologija čini mogućim kreiranje raznovrsnih aplikacija širokog polja upotrebe za sve korisnike. Različiti aspekti svakodnevnog života - kod kuće, u pokretu, u školi, na poslu, tokom raznog, izolovano, sa drugim ljudima, porodicom ili prijateljima - počinju da budu predmet dizajna tehnologija kako bi bili što kvalitetnije podržani i prošireni kao takvi. Pored toga što potrebe za unapređenjem aktivnosti utiču na pravce razvoja tehnologija i aplikacija, i nove aplikacije oblikuju i utiču na način na koje se aktivnosti obavljaju.

Dok je početkom 90-tih akcenat bio na dizajnu korisničkog interfejsa za individualne korisnike, sredinom 90-tih se javlja potreba za tehnološkom podrškom grupe korisnika koja komunicira i sarađuje u obavljanju različitih aktivnosti. Tada se javlja i razvija disciplina "kooperativni rad podržan računarima" (eng. *Computer-supported Cooperative Work, CSCW*). Informacioni sistemi (eng. *Information Systems, IS*) predstavljaju drugu oblast koja doživljava značajan razvoj u domenima kao što su poslovno okruženje, zdravstvo i edukacija. Potrebe razvoja novih aplikacija zahtevaju proširivanje i usložnjavanje timova koji su na njima radili. Nove, pridodate uloge obuhvataju stručnjake za grafički dizajn, industrijski dizajn, vizuelne medije, film, sociologe, antropologe i druge profesije. Širi skup uloga je donosio potrebnu mešavinu veština i razumevanja različitih domena aplikacija kao glavni preduslov uspešnog dizajna sa aspekta krajnjeg korisnika. Na primer, dizajn aplikacije za planiranje sa podsetnikom za porodicu zahteva razumevanje interakcije među njenim članovima. Izrada interaktivnog alata za kreiranje priča namenjenog deci podrazumeva znanje o tome kako deca pišu i doživljavaju priče. Razvoj interaktivnog vodiča za posetioce umetničke galerije mora uzeti u obzir ponašanje i kretanje korisnika u javnim prostorima.

Prva dekada 21. veka donosi značajno unapređenje hardverskih mogućnosti uređaja u pogledu procesorskih i memorijskih kapaciteta. Uporedo se povećava propusnost Internet-a kao globalne mreže za prenos i razmenu podataka između

unapređenih uređaja. Ovo dovodi do unapređenja i daljeg razvoja sistema koji podržavaju rad većeg broja korisnika u istim ili različitim vremenima kao što su javni displeji (eng. *public displays*). Pored toga, javljaju se unapređeni mobilni uređaji poput pametnih telefona (eng. *smartphone*) i prenosivih računara (eng. *notebook*). Značajan tehnološki napredak u ovom periodu predstavlja pojava dodirnih displeja (eng. *touchscreen*) za interakciju sa uređajima, pre svega u mobilnim tehnologijama. Na ovaj način je korišćenje tehnologije počelo da liči na interakciju sa objektima iz realnog sveta (eng. *direct object manipulation*). Tako je za korisnike jednostavnije da biraju i pomeraju objekte korišćenjem prstiju ruke, umesto korišćenjem miša i tastature. Sa druge strane, interakcija zahteva pažljiv dizajn kako bi bila što jednostavnija i intuitivnija za korisnike.

Konačno, druga dekada je dovela do napretka u razvoju postojećih tehnologija, kao što su virtuelna stvarnost i video igre, kao i do razvoja novih tehnologija kao što su kombinovana i proširena stvarnost (eng. *mixed and augmented reality*), Internet stvari (eng. *Internet of Things, IoT*) i konverzijski korisnički interfejsi (eng. *Conversational User Interface, CUI*). Nove i unapređenje mogućnosti tehnologije zahtevaju i nove pristupe u dizajnu kako bi bile pristupačne i upotrebljive njihovim korisnicima.

Nezavisno od domena konkretne aplikacije, korisnici vide i doživljavaju sistem primarno preko korisničkog interfejsa. GUI i dalje predstavlja najzastupljeniji tip korisničkog interfejsa i zbog toga će biti glavni predmet analize ovog udžbenika. Preduslov za dizajn upotrebljivih i atraktivnih korisničkih interfejsa jeste poznavanje i razumevanje principa koji će biti predstavljeni u nastavku, a koji određuju način na koji doživljavamo i koristimo tehnologiju.

1.3. Ciljevi dizajna Interakcije Čovek-Računar

Osnovna pretpostavka dobrog i uspešnog dizajna je razumevanje potreba ciljne grupe korisnika. Deo ovog procesa jeste jasno definisanje ciljeva budućeg sistema. Na primer, da li je cilj kreirati visoko efikasan sistem koji će povećati produktivnost korisnika u njihovom radu, ili sistem koji mora biti izazovan i motivišući kako bi podržao učenje.

U opštem slučaju, ciljeve možemo podeliti na ciljeve upotrebljivosti i ciljeve korisničkog doživljaja (eng. *user experience, UX*). Ciljevi upotrebljivosti su pragmatični po prirodi, što znači da je njihov akcenat na kvalitetu obavljanja same aktivnosti. Na primer, efikasnost, lako pamćenje ili mali broj grešaka tokom korišćenja sistema. Zbog toga se kaže i da je upotrebljivost objektivna osobina korišćenja sistema jer više fokusirana na samu aktivnost, odnosno na funkciju sistema. Sa druge strane, korisnički doživljaj je više orijentisan ka krajnjem korisniku kao složeniji i dublji psihološki odgovor na interakciju sa

sistemom. Ovo obično podrazumeva pojavu specifičnih emocija kod korisnika. Na taj način, određeni sistem doživljavamo kao izazovan, stimulišući ili estetski privlačan. Zbog toga se kaže da je korisnički doživljaj subjektivna osobina korišćenja sistema kao emocionalna reakcija koja je prirodno promenjiva među korisnicima.

Važno je razlikovati ove osobine jer one mogu, ali i ne moraju biti međusobno uslovljene. Sa jedne strane, sistem može biti visoko upotrebljiv, ali izazivati nizak doživljaj korisnika. Na primer, sistem za navigaciju u toku vožnje. Obrnuto, sistem može izazivati snažan doživljaj korisnika, ali biti složeniji za upotrebu. Na primer, igranje video igre specifičnog žanra.

Do sada smo spomenuli tri karakteristike interaktivnog računarskog sistema - korisnost, upotrebljivost i korisnički doživljaj. Relacije između ovih osobina su date na slici u nastavku.



Slika 6. Hijerarhija potreba korisnika tehnologije kao funkcionalnost (korisnost), upotrebljivost, i zadovoljstvo (doživljaj korisnika). Preuzeto i adaptirano iz [6].

U svojoj knjizi o dizajniranju proizvoda koji pružaju zadovoljstvo kod korisnika Patrik Džordan uspostavlja relacije između specifičnih ljudskih potreba [6]. Osnovna potreba koja mora biti zadovoljena jeste da proizvod mora sadržati odgovarajuće funkcije neophodne za izvršavanje aktivnosti za koje je namenjen. Skup funkcija definiše korisnost sistema. Jednom kada korisnici

budu zadovoljni funkcijama koje imaju na raspolaganju, oni žele da te funkcije budu lake za korišćenje, odnosno upotrebljiv sistem. Konačno, kada se naviknu na funkcije koje su lake za korišćenje, očekivanja korisnika rastu - oni žele proizvode koji pružaju prijatan doživljaj korisnika.

Ako govorimo o softverskoj aplikaciji, sa aspekta kreiranja tehnologije, korisnost je predmet izučavanja razvoja softvera (eng. *software development*). Upotrebljivost i korisnički doživljaj su predmet analize IČR. Fokus knjige će biti na upotrebljivosti, sa osvrtom na elemente doživljaja korisnika.

1.3.1. Ciljevi upotrebljivosti

Upotrebljivost je osobina interaktivnog sistema koja govori u kojoj meri je sistem lak za korišćenje. Opšti cilj upotrebljivosti je optimizacija korišćenja interaktivnih proizvoda u obavljanju aktivnosti na poslu, u školi i svakodnevnom životu.

Koncept upotrebljivosti je opšt i složen, te je detaljnije opisan skupom atributa kako bi bio merljiv i konkretnije definisan. Jakob Nilsen [7] je definisao pet atributa vođenih konkretnim ciljem upotrebljivosti kao:

- Lakoća učenja (eng. *learnability*) - govori koliko lako korisnici mogu koristiti sistem kada ga koriste prvi put, odnosno koliko lako se uči kako da se koristi sistem.
- Efikasnost (eng. *efficiency*) - jednom kada korisnik nauči da koristi sistem, koliko brzo može obavljati zadatke korišćenjem sistema.
- Lakoća pamćenja (eng. *memorability*) - kada nakon određenog perioda nekorišćenja korisnik koristi sistem, koliko lako se priseća da koristi sistem ili mora učiti iznova kako da ga koristi.
- Greške (eng. *errors*) - koliko grešaka korisnik pravi prilikom korišćenja, koliko su nastale greške ozbiljne i da li se sistem može nastaviti normalno koristiti nakon grešaka.
- Zadovoljstvo (eng. *satisfaction*) - govori da li je prijatno koristiti sistem.

Lakoća učenja određuje jednostavnost savladavanja funkcija sistema sa kojima se korisnik susreće prvi put, odnosno kada uči kako da koristi sistem. Poznato je da ljudi ne vole da provode puno vremena učeći kako da koriste novi uređaj ili aplikaciju. Poželjno je da korisnici ulože što manje napora i vremena u učenju korišćenja sistema i relativno brzo steknu kompetentnost nad sistemom. Ovaj atribut je važan za interaktivne proizvode za svakodnevnu upotrebu kao što su onlajn aplikacije za razonodu, kao i aplikacije koje se koriste povremeno. Primeri mogu biti videokonferencije ili sistemi telepresustva. U određenim

slučajevima, korisnici su spremni da ulože dodatni napor u učenje korišćenja složenijih sistema koji pružaju širi skup funkcionalnosti. U ovu kategoriju aplikacija mogu spadati alati za dizajn ili programi specifične namene kao što su obrada teksta, slike ili videa. U tim slučajevima dokumentacija ili tutorijali mogu biti pomoćno sredstvo za učenje pružajući interaktivne materijale prilagođene znanju korisnika sa pripremljenim vežbama. Sa druge strane, mnogi korisnici imaju poteškoća da povežu pomoćne materijale sa zadacima koje žele da obave. Ključni problem je predvideti vreme koje su korisnici spremni da investiraju u ovladavanje novim sistemom. Razvoj šireg skupa funkcionalnosti za koje većina korisnika ili nije u mogućnosti ili nije spremna da uloži vreme za učenje može uticati na usvajanje i prihvatanje novog sistema. Glavne smernice koje treba razmotriti su koliko je za korisnika jednostavno i koliko vremena zahteva učenje: (a) obavljanja ključnih zadataka u sistemu i (b) šireg skupa operacija za obavljanje većeg broja zadataka.

Efikasnost podrazumeva mehanizme pomoću kojih sistem rad korisnika čini produktivnijim, pre svega u pogledu vremena potrebnog za obavljanje zadataka. Na primer, glasovni digitalni asistent može biti efikasan u obavljanju trivijalnih zadataka kao što je preslušavanje poruka u minimalnom broju koraka ili davanju servisnih informacija o vremenu i saobraćaju. Sa druge strane, servis glasovne elektronske pošte u okviru istog asistenta može sadržati veći broj koraka za korisnika i time zahtevati više vremena za izvršenje. Opšte pravilo je da se uobičajeni i česti zadaci dizajniraju tako da minimalan broj akcija na interfejsu vodi ka izvršenju (kao što su selekcija dugmeta ili pritisak tastera tastature). Sajtovi za elektronsku kupovinu kao što je Amazon.com inicijalno od korisnika traže da unese svoje podatke, a zatim za svaku buduću kupovinu obezbeđuju takozvanu opciju jednog klika (eng. *one-click option*) za kreiranje i slanje porudžbine. Glavna smernica pri realizaciji ove osobine je da nakon što korisnici nauče da koriste sistem, treba im obezbediti mehanizme koji će samo korišćenje učiniti bržim.

Greške prilikom korišćenja sistema utiču na razvijanje osećanja sigurnosti kod korisnika (eng. *safety*) što dalje utiče na razvijanje poverenja u aplikaciju. I obrnuto, nastanak grešaka kao posledica akcija korisnika izaziva doživljaj nelagodnosti i nesigurnosti. Ovo je važan faktor koji utiče na usvajanje tehnologije od strane korisnika [8]. Cilj je što je moguće više smanjiti broj grešaka koje korisnik može napraviti. Drugim rečima, samim dizajnom interfejsa je potrebno zaštititi korisnika od kritičnih uslova po izvršavanje funkcija ili nepoželjnih situacija. Ovo su uglavnom situacije u kojima može doći do prekida toka operacije, grešaka u navigaciji i orijentaciji u aplikaciji, i gubitka podataka korisnika ili aplikacije. One obično nastaju tako što korisnik izabere pogrešnu opciju interfejsa za koju prethodno veruje da izvršava željenu akciju aplikacije. Opšti mehanizmi prevencije grešaka obuhvataju: a) jasno

izdvajanje kritičnih akcija kao što je odustajanje ili brisanje kako ne bi bile slučajno pokrenute, b) pravilan izbor simbola ikona tako da asociraju na akciju koju pokreću i c) sredstva za oporavak od grešaka ukoliko nastanu. Konkretni primeri su mehanizmi poništavanja i povrata operacija (eng. *undo/redo*) i dijalozi potvrde pre izvršavanja kritičnih operacija kao što je brisanje podataka aplikacije.

Dizajn aplikacije kao takav mora sprečiti nastanak grešaka sa ozbiljnijim posledicama po rad sistema (preventivna uloga) i omogućiti nastavak rada korisnika i nakon što greška nastane (uloga oporavka od greške).

Lakoća pamćenja se odnosi na pamćenje korišćenja sistema jednom kada se njim ovlada, odnosno nauči. Ova osobina je važna za interaktivne sisteme koji se koriste povremeno. Ako korisnici neko vreme ne koriste aplikaciju, recimo nekoliko meseci ili duže, trebalo bi da mogu da se sete ili ulože minimalan napor da se sete kako da je koriste. Korisnici ne bi trebalo da ponovo uče kako da koriste aplikaciju. Ovo se obično dešava kada su operacije u korisničkom interfejsu nejasno definisane, nelogične ili kombinovane na nelogičan način ako se sastoje iz skupa koraka. Tada korisnicima treba pomoć da se sete kako da koriste sistem. Postoji različiti načini da se ovo podrži. Na primer, korišćenjem smislenih simbola (ikona) za operacije, imena komandi i opcija menija. Zatim strukturiranje, odnosno grupisanje opcija i ikona u logičke, povezane celine (na primer, smeštanje komandi za crtanje u jedan deo interfejsa) može pomoći korisnicima da se sete gde da nađu odgovarajući alat u specifičnoj fazi zadatka ili operacije.

Glavna smernica koju treba razmotriti su kakva se podrška za pamćenje korišćenja nalazi u korisničkom interfejsu, naročito za sisteme i operacije koji se koriste povremeno.

Na osnovu prethodnog se može zaključiti da upotrebljivost kao složena osobina interaktivnog sistema varira u zavisnosti od domena i namene konkretne aplikacije. Pojedinačne osobine obrađene iznad služe istovremeno i kao vodilje u procesu dizajna aplikacije i kao kriterijumi za evaluaciju upotrebljivosti kreirane aplikacije. Iz atributa su izvedene i konkretne metrike za procenu upotrebljivosti. Na primer, vreme za izvršavanje zadatka (efikasnost), vreme za učenje zadatka (*lakoća učenja*), broj nastalih grešaka u toku izvršavanja zadatka (greške) i broj nastalih grešaka u toku više izvršavanja jednog zadatka u vremenskim razmacima (*lakoća pamćenja*).

1.3.2. Ciljevi doživljaja korisnika

Pojava novih tehnologija koje mogu uspešno podržati aktivnosti korisnika, ne samo u okviru radnog okruženja, već i deo svakodnevnice dovela je do novih zahteva korisnika. Tehnologije kao što su virtuelna stvarnost, mobilne

aplikacije i društvene aplikacije za raznovrsne domene kao što su zabava, edukacija, kućne upotrebe i javni prostori, doveli su do pojave novih ciljeva samog korišćenja. Dok je za radne tehnologije akcenat ostao primarno na efikasnosti, produktivnosti i efikasnosti korišćenja, za tehnologije svakodnevnice postaju važni ciljevi mentalne i emocionalne prirode kao što su:

- zadovoljstvo (eng. *satisfaction*),
- uživanje (eng. *enjoyment*),
- zabava (eng. *fun*),
- razonoda (eng. *entertainment*),
- motivacija (eng. *motivation*),
- estetika (eng. *aesthetics*),
- kreativnost (eng. *creativity*),
- nagrada (eng. *reward*),
- postignuće (eng. *achievement*) i drugi.

Ciljevi dizajna interaktivnih proizvoda kao što su zabava, uživanje, zadovoljstvo i estetski utisak lepote opisuju različite emocionalne doživljaje korisnika. Drugim rečima, interakcija kod korisnika izaziva različita osećanja. Doživljaj korisnika je izrazito subjektivne prirode u poređenju sa upotrebljivošću. Na primer, aplikacija za decu pomoću koje kreiraju muziku za cilj ima zabavu i razonodu, umesto efikasnosti i produktivnosti. Dakle, ciljevi korisničkog doživljaja se razlikuju od objektivnijih ciljeva upotrebljivosti u tome da oni opisuju kako korisnici doživljavaju interaktivni proizvod iz svoje perspektive, pre procene upotrebljivosti ili produktivnosti sa aspekta samog sistema. Sa konceptualnog stanovišta, može se reći da je upotrebljivost osobina više orijentisana ka tehnologiji, dok je korisnički doživljaj više na strani korisnika.

Veliki broj istraživanja u oblasti doživljaja korisnika obavlja se u industriji zabave i video igara. Ovde se otkrivaju i vrše eksperimenti sa različitim faktorima korisnika i tehnologije koji izazivaju i održavaju osećanje zadovoljstva i zabave tokom korišćenja aplikacija. Faktori uključuju pažnju, tempo (eng. *pace*), tok igre (eng. *gameplay*), interaktivnost, svesnu i nesvesnu kontroli, stil narativa (eng. *style of narrative*) i druge [9]. U zabavnim kontekstima korišćenja je čak sugerisano i da sistem može biti teži za korišćenje, ali i dalje zabavan za korisnike. Na primer, rukovanje virtuelnim objektom korišćenjem fizičkog uređaja (na primer ručni džojstik za kretanje kroz virtuelno okruženje) u poređenju sa efikasnijim mehanizmom za istu radnju (korišćenje tastera na tastaturi) može zahtevati veći napor, ali isto tako pružati prijatniji i zabavniji doživljaj korisnika.

Prepoznavanje i razumevanje relacija između ciljeva upotrebljivosti i doživljaja korisnika je veoma važno. Ciljeve je potrebno definisati pre početka dizajna i

konstantno imati u vidu u toku procesa dizajna, razvoja i evaluacije kako bi konačni proizvod u što je moguće većoj meri zadovoljavao stvarne potrebe krajnjih korisnika. Očigledno je da će različite vrste interaktivnih tehnologija imati različite potrebe upotrebljivosti i korisničkog doživljaja. Neke kombinacije ciljeva nisu kompatibilne. Na primer, može biti neizvodljivo ili nepoželjno kreirati sistem za kontrolu procesa koji je siguran i zabavan. Važno je istaći da ciljeve određuje kontekst korišćenja, aktivnosti koje treba podržati i potrebe ciljne grupe korisnika.

1.4. Upotrebljivost i principi dizajna

Jednom kada su definisani ciljevi upotrebljivosti dizajna, oni se moraju na odgovarajući način primenjivati tokom procesa dizajna. Principi dizajna su generalizovane preporuke za kritičko razmišljanje u vezi sa specifičnim aspektima dizajna. Mogu se shvatiti i kao konkretizacija ciljeva upotrebljivosti kako bi se primenjivali u toku procesa dizajna. Jedan od poznatijih, povratna informacija (eng. *feedback*), kaže da sistem mora pružiti korisnicima adekvatnu informaciju o rezultatima njihovih akcija na korisničkom interfejsu. Principi su izvedeni iz teorijskog znanja, praktičnih iskustava i zdravog razuma (eng. *common sense*). Formulirani kao preporuke, kritički sugerišu dizajnerima šta treba obezbediti, a šta izbegavati u korisničkom interfejsu. Važno je napomenuti da principi ne govore precizno kako dizajnirati konkretan interfejs (na primer, kako kreirati ikonu ili rasporediti elemente interfejsa) već više služe kao smernice da se određena pravila pravilno implementiraju u dizajnu interfejsa.

Najpoznatije i opšteprihvaćene principe je predložio i detaljno opisao Don Norman u svojoj knjizi Dizajn Svakodnevnih Stvari (eng. *The Design of Everyday Things*) [10]. Normanovi principi polaze od opštih očekivanja korisnika i daju smernice kako ispuniti očekivanja dizajnom interfejsa. U definisanju principa Norman polazi od pretpostavke da se čovek prilikom korišćenja tehnologije oslanja na znanje korišćenja objekata iz realnog sveta. Principi su formulirani kao:

- vidljivost (eng. *visibility*),
- povratna informacija (eng. *feedback*),
- ograničenja (eng. *constraints*),
- mapiranje (eng. *mapping*) i mentalni model (eng. *mental model*),
- konzistentnost (eng. *consistency*),
- priuštivost (eng. *affordances*) i metafore (eng. *metaphors*).

1.4.1. Vidljivost

Funkcije interfejsa moraju biti jasno naznačene i vidljive za korisnika. Što su vidljivije akcije i funkcije korisničkog interfejsa, veća je verovatnoća da će ih korisnici primetiti i znati kako da koriste interfejs (Slika 7). I obrnuto, što je teže pronalaženje opcija na interfejsu, korisnik će imati više poteškoća sa korišćenjem interfejsa. Norman u svojoj knjizi [10] kao pozitivan primer iz realnog sveta navodi interfejs komandne table automobila. Kontrole za različite operacije su jasno istaknute (indikator osvetljenja, promene pravca kretanja, kontrola temperature u kabini i drugi) tako da vozač ima dobar pregled opcija koje ima na raspolaganju. Relacije između rasporeda kontrola i akcija koje one pokreću generalno olakšavaju vozaču pronalazak odgovarajuće opcije kada želi da izvrši određenu radnju nad vozilom.



Slika 7. Primer vidljivosti - Ribon (eng. *ribbon*) alata *Microsoft PowerPoint* predstavlja skup povezanih panela sa preglednim i jasno vidljivom opcijama za formatiranje teksta.

1.4.2. Povratna informacija

Princip povratne informacije dolazi iz analogije sa situacijama iz realnog sveta u kojima akcija daje neposredno vidljiv rezultat, kao što su crtanje, pisanje, ili sviranje instrumenta. U svim situacijama korisnik odmah vidi napisano i nacrtano, ili čuje odsviranu melodiju tako da može nastaviti sa aktivnošću. Očekivanja su identična i kod korišćenja tehnologije, odnosno korisničkog interfejsa.

Princip ima dve važne implikacije - postojanje povratne informacije i njena istovremenost sa povezanom akcijom. U korisničkom interfejsu, svaka akcija mora imati povratnu informaciju o rezultatima njenog izvršenja, kako bi korisnik znao kako da nastavi sa korišćenjem interfejsa.

U korisničkim interfejsima se primenjuju različiti tipovi povratnih informacija na nivou kontrola kao što su dugme, ikona, stavka menija, hiperveza i drugi. Informacija može biti vizuelna, zvučna, verbalna, taktilna ili kombinacija prethodnih. Na primer, pritiskom na dugme se menja boja dugmeta (statička vizuelna informacija) ili primenjuje animacija kao što je efekat talasa (dinamička vizuelna informacija, Slika 8). Kod nekih pametnih telefona, pritisak na dugme telefona ili ikonu dodirnog displeja prouzrokuje osećaj

vibracije uređaja (taktilna informacija). Nekada klik dugmeta prouzrokuje zvučni signal (zvučna informacija). Svrha povratne informacija kod aktivacije dugmeta jeste da korisnik bude svestan da je kliknuo na dugme i pokrenuo određenu akciju na interfejsu. Važno je i da informacija bude pravovremena, odnosno da se pojavi neposredno nakon klika dugmeta kako korisnik ne bi u međuvremenu pokretao druge akcije. Odluka o vrsti i kombinacijama povratne informacije zavisi od prirode akcije koja se pokreće. Opšte pravilo je da ukoliko je akcija kritična i značajnija po izvršavanje aplikacije, onda i povratna informacija mora biti istaknutija. Obrnuto, ukoliko je u pitanju rutinska, često korišćena akcija tada povratna informacija treba biti suptilnija i diskretnija kako ne bi ometala rad korisnika.



Slika 8. Povratna informacija - Klikom na dugme kreira se vizuelni efekat kružnog talasa (eng. *ripple effect*).

Izvor: <https://material.angular.io/components/button/examples>.

1.4.3. Ograničenja

Ograničenje kao koncept dizajna se odnosi na definisanje dozvoljene interakcije sa sistemom na datom mestu u korisničkom interfejsu u datom trenutku. Na ovaj način se smanjuje mogućnost nastanka grešaka i generalno povećava efikasnost korišćenja. Postoje različiti mehanizmi za definisanje ograničenja korišćenja u interfejsu. Uobičajena praksa kod GKI je deaktivacija stavki menija kako bi korisnik u datoj fazi aktivnosti na raspolaganju imao samo dozvoljene opcije.

Edit	View	Insert	Format
Undo Crop Picture			⌘Z
Redo Clear			⌘Y
Cut			⌘X
Copy			⌘C
Copy to Scrapbook			⌘⇧C
Paste			⌘V
Paste Special...			⌘⇧V
Clear			
Select All			⌘A
Duplicate			⌘D
Delete Slide			
Remove Section			
Rename Section			

Slika 9. Primer ograničenja - Dostupne stavke menija za ažuriranje teksta alata *Microsoft PowerPoint* u trenutku kada nije selektovan tekst za obradu.

Ovakav tip ograničenja prvenstveno doprinosi prevenciji grešaka korisnika. Upotreba različitih grafičkih prezentacija može uticati na razumevanja problema ili skupa informacija. Na primer, dijagrami tokova liustruju pravlinu povezanost objekata i tokove informacija.

Norman [10] predlaže tri tipa ograničenja: fizičko, logičko i kulturološko. Fizičko ograničenje definiše način na koji fizički objekti svojim izgledom i oblicima ograničavaju rukovanje predmetima. Na primer, prenosiva memorija ima takav priključak da odgovara specifičnim ulazima na računaru. Logičko ograničenje proističe iz korisnikovog razumevanja načina na koji funkcionišu pojave u realnom svetu. Ono se oslanja na ljudsko rezonovanje akcija i njihovih posledica. Dizajn koji akcije i njihove rezultate čini očiglednim pospešuje logičko zaključivanje kod korisnika.



Slika 10. Primer ograničenja - Logičan raspored kontrola za slušanje muzike alata *iTunes*.

Kulturološka ograničenja potiču iz usvojenih dogovora, običaja, navika ili obrazaca ponašanja. Na primer upotreba crvene boje kao signala upozorenja, upotreba određenih audio signala za opasnost, i upotreba simbola smeška (eng. *smiley*) za emociju sreće. Ovakva ograničenja su apstraktne prirode i rezultat praksi korisnika. Jednom kada ih nauči i prihvati kulturološka grupa, oni postaju univerzalno prihvaćene konvencije. Dve univerzalne konvencije za GKI su upotreba prozora za prikaz sadržaja i ikona za operacije i dokumente.

1.4.4. Mapiranje, mentalni model i metafore

Mapiranje se odnosi na relacije između kontrola korisničkog interfejsa i njihovih efekata u stvarnosti. Primer dobrog mapiranja između kontrola i njihovih efekata predstavljaju tasteri-strelice za navigaciju na tastaturi (gore, dole, levo, desno) za reprezentaciju kretanja kursora. Mapiranje relativne pozicije kontrola i efekata je takođe važno. Programi za slušanje muzike obično prate istovetan raspored dugmadi sa plej (eng. *play*) dugmetom u sredini, dugmetom za vraćanje nazad (eng. *rewind*) na levoj strani i dugmetom za kretanje napred (eng. *forward*) na desnoj strani posmatrača (Slika 10). Promena redosleda dugmadi bi izazvala probleme u korišćenju zbog toga što nije u skladu sa onim na šta su korisnici navikli.

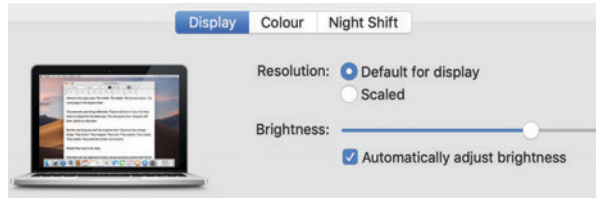
Opštiji, povezani pojam je mentalni model (eng. *mental model*) koji predstavlja mentalnu reprezentaciju o tome kako nešto funkcioniše. Ljudi imaju predefinisane mentalne modele iz realnog sveta i prethodnog iskustva korišćenja kojima se vode kada koriste tehnologiju.

Metafore (eng. *metaphors*) u dizajnu tehnologije podrazumevaju transfer između objekata i relacija između njih iz izvornog domena u ciljne objekte i relacije u odredišnom domenu [10].

U dizajnu korisničkog interfejsa izvorni skup predstavlja naučeni objekti i radnje iz svakodnevnog života koji se preslikavaju u objekte i akcije korisničkog interfejsa. Najbolji primer korišćenja metafore jesu simboli ikona. Za pravilnu upotrebu ikona je ključan odabir simbola tako da asociraju na akciju koja se pokreće klikom na ikonu. Na primer, simbol štampača asocira na akciju štampe, dok je simbol makaza povezan sa operacijom isecanja (eng. *cut*). Ikone predstavljaju jedan od najzastupljenijih elemenata modernih interfejsa. U slučaju manjeg skupa uobičajenih operacija mogu se koristiti standardni, opštepoznati simboli. Međutim specifičnije operacije zahtevaju pažljiv izbor simbola za prezentaciju akcije. Preporuka je i koristiti standardizovane kolekcije simbola za ikone koji su rezultat višedecenijske prakse dizajna korisničkih interfejsa kao što je *Google Material Design* [11].



(a)



(b)

Slika 11. Primer metafore. Klizač fizičkog uređaja (a) predstavlja metaforu iz realnog sveta koja se koristi u dizajnu korisničkog interfejsa (b).

1.4.5. Konzistentnost

Konzistentnost podrazumeva dizajn korisničkih interfejsa koji imaju slične operacije i koriste slične elemente za izvršavanje sličnih zadataka. Konzistentan interfejs implementira pravila konzistentnosti kao što su korišćenje identičnih simbola ikona za jednu akciju na različitim mestima u interfejsu. Na primer, konzistentna operacija koristi istu ulaznu akciju za selektovanje objekata na interfejsu, kao što je klik na levo dugme miša. Nekonzistentni korisnički interfejsi omogućavaju odstupanje od usvojenih pravila korišćenja. Na primer, korišćenje desnog dugmeta miša za selekciju objekata interfejsa ili levog dugmeta miša za otvaranje padajućeg menija za dodatne operacije sa objektom. Nekonzistentnost u dizajnu korisničkog interfejsa po pravilu povećava verovatnoću grešaka koje korisnik pravi u toku korišćenja.

Osnovna prednost konzistentnih korisničkih interfejsa je da su lakši za učenje i korišćenje. Korisnici jednom uče standardne načine korišćenja interfejsa kao operacije nad objektima interfejsa. Princip se lakše primenjuje na jednostavnije korisničke interfejse, sa ograničenim skupom operacija. Sa povećanjem broja operacija teže je i pronaći standardne načine za njihovo prezentovanje u interfejsu. Rešenje je grupisanje operacija u logičke celine ili skupove na osnovu sličnosti. Na primer, aplikacije za u obradu teksta grupišu komande za rad sa specifičnim vrstama sadržaja koje se mogu pojaviti u tekstu. Tako Fajl (eng. *file*) meni sadrži operacije za rad sa dokumentom kao celinom (otvaranje, čuvanje, zatvaranje i druge). Slično tome, opcije za uređivanje teksta se nalaze u Edit (eng. *edit*) meniju. Izbor opcije se svodi na izbor nadređene kategorije (menija) i vizuelno skeniranje stavki dok se ne uoči odgovarajuća opcija. Poželjno je da se iste opcije nalaze u istoimenim menijima u različitim verzijama aplikacije ili u aplikacijama slične namene.

1.4.6. Priuštivost

Priuštivost (eng. *affordance*) je termin koja označava osobinu objekta koja neposredno govori korisnicima kako da koriste objekat. Na primer, fizički dizajn dugmeta miša govori korisniku da treba da pritisne dugme. Može se posmatrati i kao davanje tragova (eng. *clues*) o načinu korišćenja objekta. Objekat naravno može biti materijalan (miš) ili digitalan (opcija interfejsa). Kada su ove osobine fizičkog objekta istaknute može se lakše naučiti kako da se koristi. Na primer, kvaka na vratima govori da je treba pritisnuti ili ručica šolje za kafu da je treba uhvatiti [10].

Norman je uveo ovaj koncept kasnih 80-tih godina 20. veka u svojim diskusijama o dizajnu svakodnevnih objekata [10]. Od tada se koristi u dizajnu korisničkog interfejsa, odnosno objekata interfejsa kako bi oni svojim izgledom korisniku “govorili” kako da ih koristi. Na primer, elementi kao što su dugmad, ikone, veze (eng. *links*) i trake za skrolovanje (eng. *scrollbars*) zahtevaju pažljiv dizajn kako bi ikone, veze i dugmad pozivali na klik, a trake na pritisak i pomeranje ručica (eng. *handles*). Koncept je apstraktne prirode i zahteva pažljivo razmatranje i primenu u dizajnu korisničkih interfejsa.

2. Tipovi i stilovi interakcije

Dizajn interaktivnog sistema polazi od aktivnosti korisnika koje treba da podrži. Skup aktivnosti opisuje korisničke zahteve za datim sistemom. Korisnički zahtevi definišu prostor problema koji sistem rešava (eng. *problem space*). Sa druge strane, za korisnika je važan i način na koji će ove aktivnosti obavljati korišćenjem budućeg sistema. Ova očekivanja korisnika se opisuju konceptualnim modelom (eng. *conceptual model*) koji se definiše kao:

“Opis predloženog sistema kao integrisanog skupa ideja, koncepata, ciljeva, izgleda i ponašanja na način razumljiv krajnjim korisnicima.”
[12]

Ključni momenat u procesu dizajna jeste definisanje ciljeva korisnika koji određuju zadatke koji će se obavljati korišćenjem proizvoda. Na primer, da li će prvenstveno vršiti pretragu informacija, kreiranje dokumenata, komunikaciju sa drugim korisnicima, evidenciju događaja ili neku drugu aktivnost.

Ovakvo razmatranje zahteva razdvajanje tipa i stila interakcije. Tip interakcije je određen njenim krajnjim ciljem, dok stil definiše način na koji se ona odvija. Primeri tipa interakcije bi bili pretraživanje ili direktno postavljanje pitanja sistemu na maternjem jeziku. Za zadati, utvrđeni tip stilovi bi mogli biti sistem menija, glasovne komande ili tekstualne komande. Tip interakcije je pojam apstraktniji od stila interakcije. Dok tip opisuje prirodu aktivnosti korisnika, stil se odnosi na konkretnu implementaciju aktivnosti, odnosno specifične elemente korisničkog interfejsa.

Jednom kada se definišu aktivnosti i njihovi tipovi interakcije, dalje se konkretizuju elementi korisničkog interfejsa izborom detalja izgleda i ponašanja (eng. *look and feel*). Ovaj proces se odigrava u iteracijama, korišćenjem različitih metoda. Metode obuhvataju skice interfejsa (eng. *sketching*), vizuelne narative interfejsa (eng. *storyboarding*), opise scenarija i prototipiziranje izgleda i ponašanja korisničkog interfejsa (eng. *prototyping*).

2.1. Tipovi interakcije

Definišu konceptualne modele vođene aktivnostima. Tipovi interakcije ili aktivnosti prilikom interakcije sa sistemom mogu biti:

- instrukcije,
- konverzacija,
- manipulacija i navigacija,
- eksploracija i pretraživanje.

Važno je napomenuti da tipovi interakcije nisu međusobno isključivi i mogu se kombinovati. Tako je moguće da korisnik zadaje instrukcije u toku konverzacije ili vrši navigaciju dok pretražuje sadržaje interfejsa. Međutim, svaki tip ima različite karakteristike i diktira specifične implementacije interfejsa u smislu pogodnosti određenih stilova interakcije.

Prvi tip je vođen principom da korisnik zadaje instrukcije sistemu dok obavlja zadatke. Ovo se postiže različitim stilovima kao što su kucanje komandi, izbor stavki menija (korišćenjem miša ili dodirnog displeja), izgovaranjem komandi, selekcijom dugmadi ili korišćenjem funkcijskih tastera tastature.

Drugi tip je zasnovan na konverzaciji korisnika sa sistemom poput razgovora sa drugim korisnikom. Korisnik razgovara sa sistemom ili zadaje tekstualna pitanja na koja sistem odgovara putem teksta ili glasa.

Treći tip omogućava korisnicima da upravljaju digitalnim objektima interfejsa i da se kreću kroz interfejs. Pretpostavka je da virtuelno okruženje u određenoj meri podseća na realan svet i na taj način omogućava korisnicima da koriste znanje o interakciji sa fizičkim objektima dok interaguju sa virtuelnim objektima.

Četvrti tip pretpostavlja interfejs čiji je sadržaj organizovan tako da korisnik pretražuje i pronalazi informacije, ili uči bez eksplicitnog zadavanja pitanja sistemu.

U nastavku će svaki od tipova biti detaljnije opisan.

2.1.1. Instrukcije

Ovaj tip interakcije opisuje aktivnosti u kojima korisnik zadaje komande sistemu. Primeri uključuju informaciju o vremenu, štampanje datoteke, ili definisanje podsetnika za aktivnosti korisnika. Dizajn mnogih ulaznih uređaja je prilagođen ovom modelu interakcije, kao što je tastatura ili pametni mikrofoni i zvučnici virtuelnog asistenta za glasovnu komunikaciju (Slika 12).



Slika 12. *Amazon Alexa* pametni zvučnici i mikrofoni za zadavanje glasovnih komandi virtuelnom asistentu. Izvor: Raysonho @ Open Grid Scheduler / Scalable Grid Engine CC0.

Načini zadavanja instrukcija su raznovrsni, od korišćenja dugmadi do unosa teksta. Prvi operativni sistemi kao što su UNIX i DOS dizajnirani su prvenstveno kao komandni sistemi kojima su korisnici zadavali tekstualne naredbe specifičnih formata. Kasnije, u vizuelnim operativnim sistemima zasnovanim na prozorima, zadavanje komandi je olakšano korišćenjem prečica tastature, stavki menija ili pokazivačkih uređaja (miš). Primeri komandnih aplikacija su programi za obradu teksta, klijenti elektronske pošte i alati za dizajn podržan računarima (eng. *Computer Aided Design*, CAD). Komande su obično predstavljene ikonama i grupisane radi lakšeg prepoznavanja i korišćenja interfejsa (Slika 13).



Slika 13. Primer vizuelnog komandnog interfejsa - *Adobe Illustrator*, alat za obradu slika, ima korisnički interfejs sastavljen od tematskih grupa alata kao ikona za instrukcije koje predstavljaju specifične operacije nad vizuelnim sadržajem.

Jedna od osnovnih prednosti komandnog konceptualnog modela je da podržava efikasnost korišćenja interfejsa. Ovaj tip interakcije je dobar izbor za česte operacije nad jednim ili većim brojem objekata. Primer je čuvanje, brisanje ili premeštanje poruka elektronske pošte ili datoteka.

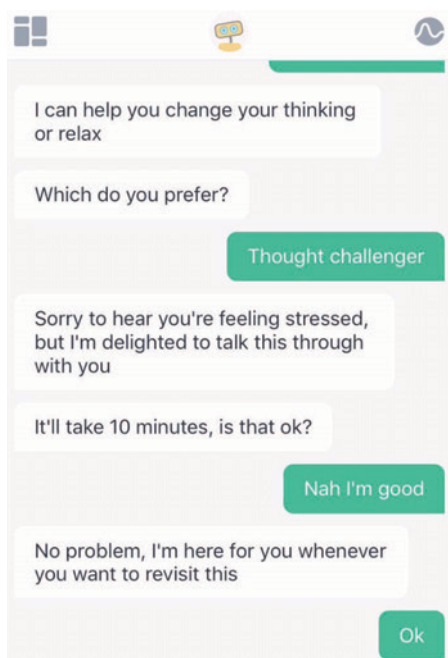
Velika pažnja se posvećuje unapređenju upotrebljivosti komandnih korisničkih interfejsa. Primeri obuhvataju izgled komandi (korišćenje skraćenica, izbor reči i simbola), kompozicija (kombinovanje različitih komandi) i organizacija (struktura komandi kao što su izgled i raspored stavki u meniju).

2.1.2. Konverzacija

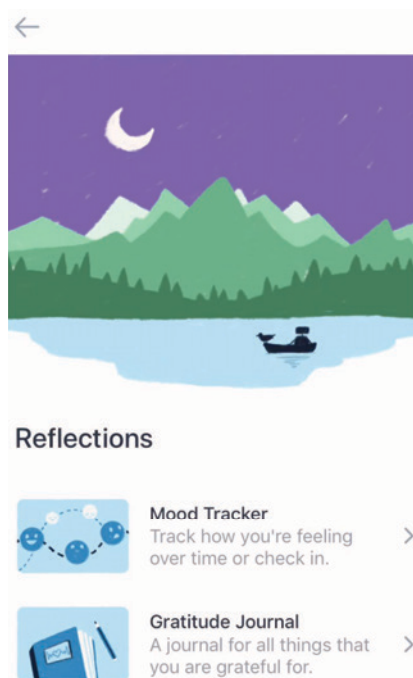
Ovaj tip interakcije je zasnovan na metafori svakodnevne međuljudske konverzacije, gde je sistem partner u komunikaciji. Na ovaj način, sistem je kreiran tako da oponaša čoveka u konverzaciji tako što vodi dijalog sa krajnjim korisnikom. Za razliku od prethodnog tipa koji implementira jednosmernu komunikaciju korisnik-sistem, ovde je komunikacija dvosmerna u smislu postojanja dijaloga umesto izvršavanja instrukcija koje zadaje korisnik. Ovaj tip interakcije je pogodan za scenarije u kojima korisnik traži specifične informacije ili vodi diskusije na različite teme.

Razvoj mašinskog učenja (eng. *machine learning*, *ML*) i procesiranja prirodnog jezika (eng. *natural language processing*, *NLP*) doveo je do pojave

intelligentnih konverzacijskih agenata poznatijim pod nazivom četbot (eng. *chatbot*). Komercijalni sistemi, kao što su Amazon Alexa i Google Assistant, koriste glasovnu konverzaciju i podržavaju svakodnevne aktivnosti korisnika kao što su dobijanje servisnih informacija, planiranje ličnih aktivnosti i naručivanje. Pored toga, agenti su našli i široku primenu u zdravstvu [13] (Slika 14).



(a)



(b)

Slika 14. Primer konverzacije - Woebot¹ je virtuelni konverzacijski agent koji ima ulogu mentalnog terapeuta. Interakcija je zasnovana na tekstualnom dijalogu sa korisnikom (a) i podržana dodatnim funkcionalnostima (b).

Glavna prednost ovog tipa interakcije je da omogućava korisnicima, naročito onima sa malo ili bez iskustva, korišćenje sistema na način koji im je već poznat, na osnovu znanja vođenja razgovora sa drugim ljudima. Ovo istovremeno može biti i ograničenje u slučaju kad sistem ne razume pitanje korisnika zbog nemogućnosti ispravne interpretacije. Ovo je posledica nemogućnosti pozadinskih algoritama da ispravno interpretiraju značenje

¹ Woebot - virtuelni agent za mentalnu terapiju: <https://woebothealth.com/the-app/> (15.11.2020).

složenijih rečenica kako bi prepoznali pravu nameru korisnika. Ovde spadaju situacije koje zahtevaju analizu konteksta rečenica kako bi se zaključilo pravo značenje reči. Dok ljudi ovakve situacije rešavaju jednostavno u svakodnevnom govoru, ML i NLP algoritmi još uvek mogu imati poteškoća u ovim slučajevima.

Ovaj problem se donekle pokušava ublažiti vizuelnim dizajnom agenata. Često se predstavljaju kao različiti tipovi karaktera, od stvarnih ljudskih figura to animiranih virtuelnih likova. Na ovaj način korisnik doživljava sistem kao vidljivog i opipljivog partnera u dijalogu nalik na čoveka, umesto da samo čuje šta sistem govori ili čita što sistem napiše. Pored toga, agenti pokazuju i neke osobine svojstvene čoveku, kao što su humor, sreća, zadovoljstvo, oduševljenje i druge. Međutim, i ovaj pristup zahteva oprez zbog toga što se može desiti da kada sistem pogrešno interpretira pitanje, nezadovoljstvo korisnika bude veće zbog većih očekivanja koja proističu iz njihove percepcije da razgovaraju sa drugom osobom [14].

Konverzacija sama po sebi predstavlja intuitivan tip interakcije ali zahteva pažljivo razmatranje prirode aktivnosti sa aspekta složenosti dijaloga i načina prezentacije korisničkog interfejsa (tekstualni, glasovni, vizuelni ili kombinacija prethodnih).

2.1.3. Manipulacija i navigacija

Tip interakcije koji podrazumeva upravljanje objektima i kretanje kroz virtuelno okruženje oslanjajući se na način na koji korisnici obavljaju slične aktivnosti u stvarnom svetu. Tako se virtuelni objekti mogu birati, pomerati, otvarati, zatvarati i skalirati (eng. *zoom*). Neke od ovih akcija, poput skaliranja, transformacije objekata i navigacije kroz virtuelni prostor, predstavljaju proširenje u odnosu na ono što je moguće u realnosti.

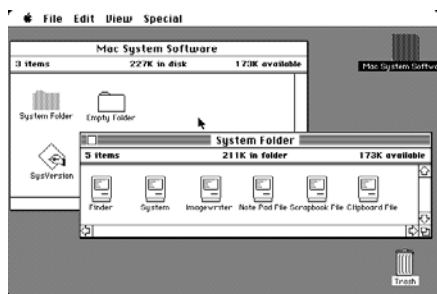
Ovaj model je prvi predstavio i opisao Ben Šnajderman 1982. godine [15] kao interfejs direktne manipulacije (eng. *direct-manipulation interface*). Ova vrsta interfejsa se odlikuje sledećim karakteristikama:

- stalna vidljivost objekata i ciljnih akcija,
- reverzibilne i inkrementalne akcije sa trenutnom povratnom informacijom za korisnika,
- akcije fizičkog pokreta i pritiska dugmeta umesto zadavanja komandi složene sintakse.

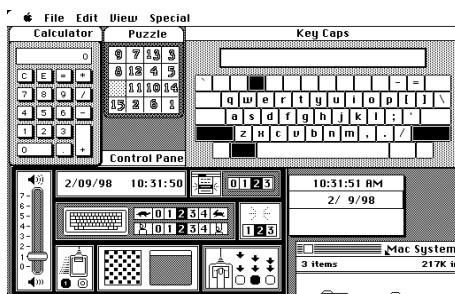
Prednosti direktne manipulacije uključuju:

- brzo učenje osnovnih funkcionalnosti od strane početnika,
- brže korišćenje od strane iskusnih korisnika,
- lakše pamćenje operacija od strane povremenih korisnika,
- smanjena potreba za obaveštenjima o greškama (korisnici neposredno vide da li je nastala greška i koriguju svoje akcije),
- niži nivo nezadovoljstva kod korisnika,
- veći nivo samopouzdanja i doživljaja kontrole kod korisnika.

Apple Computer je među prvim kompanijama koja je kreirala operativni sistem kod kojeg je direktna manipulacija najzastupljeniji tip interakcije (Slika 15). Kako bi iskoristili predstave koje ljudi formiraju na osnovu manipulacije objektima u fizičkom svetu, dizajn interfejsa u što je moguće većoj meri pokušava da simulira kontekst realnosti.



(a)



(b)

Slika 15. Primer direktne manipulacije - *Apple* operativni sistem iz 1984. godine je koristio sisteme prozora i omogućavao prevlačenje objekata korisničkog interfejsa. Izvor: Andy F. Mesa / The Early Mac OS <http://applemuseum.bott.org/sections/os.html> (15.11.2020).

Video igre, alati za vizuelizaciju podataka i virtuelna stvarnost predstavljaju primere interfejsa direktne manipulacije. Virtuelna stvarnost generiše trodimenzionalno veštačko okruženje kroz koje se korisnik kreće i čijim objektima upravlja pomoću specijalnih uređaja (Slika 16).



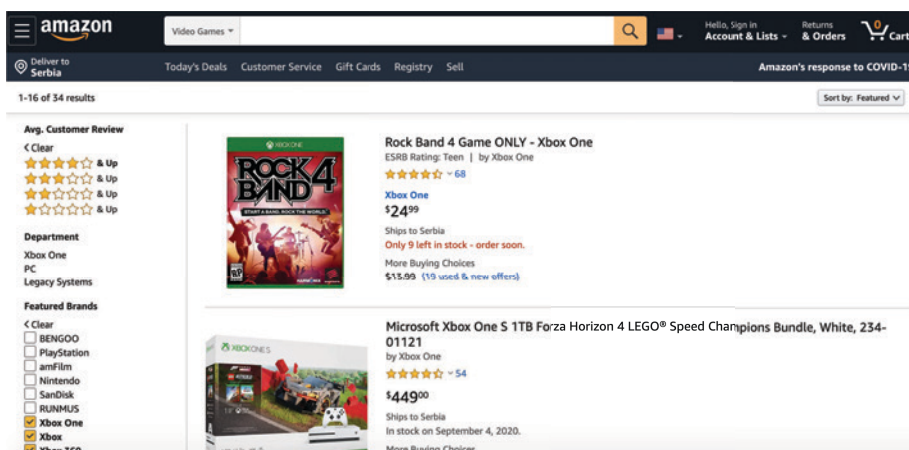
Slika 16. Virtualna stvarnost je zasnovana na manipulaciji i navigaciji kao tipu interakcije. Izvor: European Space Agency, Reality Check, CC BY-SA 3.0 IGO.

Dok sa jedne strane direktna manipulacija olakšava korišćenje interfejsa i pruža slobodu u interakciji, postoje i kritični elementi ovog tipa interakcije. Uvek je potrebno razmotriti da li je konkretan zadatak po svojoj prirodi pogodan za realizaciju primenom direktne manipulacije. Neki zadaci se teško mogu predstaviti kao objekti i neke akcije mogu biti složene za realizaciju direktnom manipulacijom. Takvi zadaci se brže i lakše mogu obaviti zadavanjem instrukcija i tekstualnim opisima objekata umesto simbola (ikone). Na primer, predstavljanje poruka elektronske pošte kao ikona sa dodatnim informacijama kao što su pošiljalac, vreme slanja i naslov bi zauzele dosta prostora na ekranu i direktna manipulacija bi značajno otežavala upravljanje porukama.

2.1.4. Eksploracija i pretraživanje

Ovaj konceptualni model je fokusiran na istraživanju i pretrazi informacija. Model se oslanja na znanje korisnika u pretrazi različitih tipova sadržaja u stvarnom svetu poput knjiga, novina, televizije, radija, biblioteka, kataloga i drugih. Ono što je zajedničko za svaku pretragu sadržaja je da korisnici vizuelno skeniraju dostupne informacije u potrazi za konkretnom informacijom

koja je predmet pretrage. Primeri aplikacija koje koriste ovaj tip interakcije su Veb sajtovi, portali, edukativni sajtovi i sajtovi za elektronsku trgovinu (Slika 17). Akcenat je na struktuiranju informacija tako da navigacija bude što je moguće efikasnija, omogućavajući korisnicima da pretražuju, pregledaju i nalaze različite vrste informacija.



Slika 17. Primer eksploracije i pretraživanja - Amazon sajt za kupovinu omogućava efikasnu pretragu proizvoda po različitim kriterijumima i njihov pregled.

2.2. Stilovi interakcije

Stil interakcije predstavlja konkretan način na koji korisnik interaguje sa sistemom. Na taj način stil definiše vrstu interfejsa koji implementira specifične tipove interakcije. Ovde se prvenstveno misli na upotrebu konkretnih elemenata korisničkih interfejsa koje diktira izabrani stil interakcije. Stilovi interakcije se mogu posmatrati i kao kanali razmene informacija sa korisničkim interfejsom koji angažuju specifična ljudska čula i aparate. Na ovaj način svaki stil koristi karakteristične uređaje za interakciju sa sistemom. Tako se stilovi mogu klasifikovati kao:

- Kucanje (eng. *typing*),
- Pokazivanje (eng. *pointing*),
- Dodir (eng. *touch*),
- Govor (eng. *speech*),
- Oset (eng. *haptics*),
- Kombinovani kao integrisani skup stilova iznad (eng. *multimodal*).

Kucanje je stil interakcije koji podrazumeva komunikaciju sa sistemom putem unosa alfanumeričkog sadržaja korišćenjem tastature. Predstavlja prvi stil interakcije koji se koristio u komandnim interfejsima računara. Kao takav i danas je zastupljen u širem spektru korisničkih interfejsa i to u kombinaciji sa drugim stilovima. Pravilo je da se koristi u onim operacijama u kojima je komplikovanija ili nije izvodljiva primena drugih stilova. Na primer, pretraga po različitim ključnim rečima koje mogu biti odvojene ili kombinovane po frazama. Elementi korisničkog interfejsa koji realizuju ovaj stil su različite vrste tekstualnih polja.

Pokazivanje predstavlja stil u kojem se operacije izvršavaju izborom odgovarajućih opcija korišćenjem pokazivačkih uređaja (miš ili panel osetljiv na dodir). Predstavljaju efikasniji i jednostavniji način interakcije sa sistemom tako što se komande zadaju jednim pokretom, odnosno selekcijom odgovarajuće opcije. Prvi put se javlja sa pojavom GKI i danas predstavlja jedan od najzastupljenijih stilova interakcije u korisničkim interfejsima. Stil realizuju elementi GKI kao što su dugmad, meniji, hiperveze, ikone i trake za navigaciju.

Dodir je stil interakcije koji se oslanja na čulo dodira. Ovde prvenstveno mislimo na dodirne korisničke interfejse (eng. *touchscreen user interfaces*). Važno je napomenuti da je u ovom slučaju interakcija sa interfejsom direktna - korisnik prstima ruke selektuje ili prevlači objekte na interfejsu. Kod pokazivanja, interakcija se ostvaruje indirektno, posredstvom uređaja - klikom na dugme miša ili pokretima na panelu osetljivom na dodir. Iako krajnji rezultat može biti identičan, interakcija je drugačija sa aspekta krajnjeg korisnika. Dodir je podrazumevani stil interakcije kod mobilnih uređaja. Ekрани osetljivi na dodir detektuju prisustvo i lokaciju dodira korisnika kako bi se prepoznala i izvršila željena akcija. Glavna prednost je lako učenje korišćenja interfejsa. Tek sa pojavom dodirne interakcije princip direktne manipulacije je zapravo implementiran kao takav. Elementi korisničkog interfejsa su kao kod kucanja i pokazivanja, sa tim što se biraju i aktiviraju dodirrom.



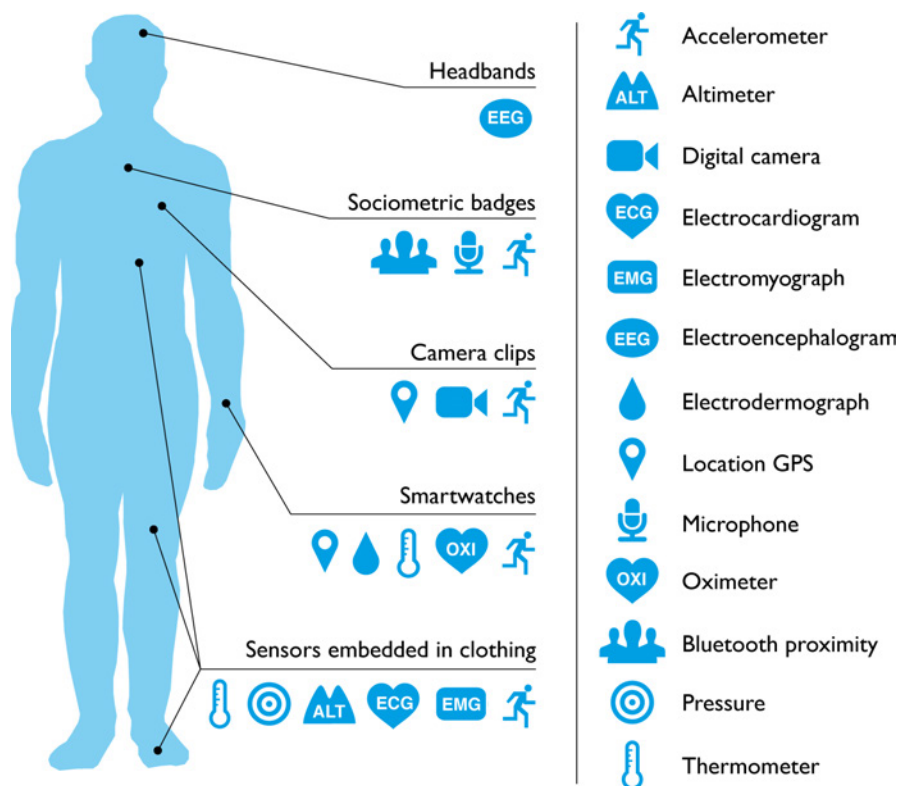
Slika 18. Dodir kao stil interakcije - Interaktivna tabla u učionici kao dodirni korisnički interfejs. Izvor: By Geofflambeth - Own work / CC BY-SA 4.0.

Govor kao stil pokušava da interakciju korisnika i računara još više približi svakodnevnoj međuljudskoj komunikaciju. Korisnik govori interfejsu koji procesira izgovorene reči kako bi identifikovao nameru korisnika i izvršio odgovarajuću akciju. Nakon izvršenja akcije sistem izgovara korisniku rezultat. Na ovaj način, interakcija sa sistemom poprima oblik dijaloga. Iako su se govorni interfejsi pojavili još 80-tih godina 20. veka, do značajnijeg napretka tehnologija za interpretaciju, upravljanje i generisanje prirodnog govora dolazi u drugoj dekadi 21. veka. Ovaj napredak je pre svega posledica značajnijeg razvoja NLP i ML oblasti. Ovo je dovelo do pojave četbot tehnologije koja nalazi sve veću primenu u različitim oblastima.

Centralni mehanizam četbota je upravljanje dijalogom (eng. *dialog management*) [16]. Sistemi zasnovani na pravilima (eng. *rule-based systems*) traže odgovore na pitanja korisnika iz postojećeg skupa (baza znanja), koji su organizovani i pretražuju se po principu hijerarhije stabla. Ovakvi sistemi su otporniji na greške u dijalogu sa korisnikom, ali su pogodni za ograničene domene pitanja i odgovora. Sa druge strane, statistički sistemi dijaoga (eng. *statistical systems*) generišu odgovore na pitanja korisnika iz šireg skupa podataka, primenom različitih ML metoda. Ovi sistemi su efikasniji i efektivniji za realizaciju složenijih dijaloga, kod pitanja korisnika koja je potrebno interpretirati iz konteksta celokupnog razgovora. Pored toga, još uvek imaju

ograničenu preciznost i manje su otporni na greške. Hibridni pristupi kombinuju prethodna dva kako bi dijalozi sa korisnicima bili što optimalniji u pogledu sadržaja, preciznosti i pravovremenosti odgovora. Vremenom sistem kroz dijalog uči o korisniku (interesovanja, navike, potrebe, stavovi i druge informacije) tako da počinje da preuzima inicijativu i daje preciznije odgovore. Sistemi za upravljanje dijalozima u najvećoj meri koriste dostupne podatke na engleskom jeziku [17]. Ovo za posledicu ima to da su drugi jezici nešto slabije podržani u glasovnim korisničkim interfejsima u ovom trenutku.

Oset (eng. *haptics*) je stil interakcije koji je zasnovan na percepciji dodira kao osobini ljudskog tela. Stil oseta ima šire i drugačije značenje od stila dodira koji se odnosi na direktan dodir interfejsa prstima ruke. Sastoji se u primeni vibracija i mehaničkih sila na telo korisnika koristeći senzorske uređaje postavljene na odeću, ili koji su deo uređaja koje oni nose (pametni telefon). Primeri predstavljaju pametne narukvice i satovi (eng. *smart wristband and watches*) (Slika 19). Koriste se u kombinaciji sa aplikacijom koja ima grafički korisnički interfejs preko kojeg korisnik može pratiti fiziološke i psihološke parametre koji se prikupljaju, analiziraju i prezentuju. Primene obuhvataju mentalno zdravlje (ublažavanje stresa), praćenje sna, održavanje krvnog pritiska, održavanje fizičke kondicije, praćenje težine, fizička aktivnost, i praćenje opšteg zdravlja kroz monitoring disanja, telesne temperature i rada srca.



Slika 19. Primene i senzori za realizaciju oseta kao stila interakcije. Preuzeto iz [18].

Oset kao stil interakcije se koristi u dizajnu za osobe sa posebnim potrebama (eng. *inclusive design*). Na primer, kao podrška u kretanju starijim osobama sa smetnjama vida koristi se narukvica koja vibrira u slučaju kritičnih promena pravca kretanja i robotička šetalica kao podrška hoda (eng. *robotic walker*) [19], [20].

Posebna vrsta korisničkih interfejsa zasnovanih na osetu su takozvani osetni korisnički interfejsi (eng. *tangible user interface, TUI*). Ovi interfejsi koriste čulo dodira tako što korisnik interaguje sa interfejsom preko objekata koji imaju fizičku, trodimenzionalnu formu. Dakle interfejs kao takav nije planaran, odnosno ravan već sadrži oblike iz realnosti. Za izradu ovakvih interfejsa se koriste posebne tehnologije i digitalni materijali (Slika 20).



Slika 20. Osetni korisnički interfejs za kreiranje muzičkih melodija kroz igru.
Izvor: By Daniel Williams from NYC, USA - The Reactable / CC BY-SA 2.0.

Multimodalni korisnički interfejsi (eng. *multimodal user interfaces*) integrišu veći broj kanala komunikacije čoveka i računara (modaliteta) u jednom korisničkom interfejsu. Ovi interfejsi kombinuju veći broj stilova, na primer pokazivanje i glas ili pokazivanje i dodir. Kao takvi pružaju bogatiji doživljaj korisnika korišćenjem različitih načina prenosa informacija, veću fleksibilnost u korišćenju i mogu povećati efikasnost. Sa druge strane zahtevaju pažljiv dizajn i kombinovanje modaliteta kako korisnici ne bi bili preopterećeni informacijama (eng. *information overload*). Multimodalni korisnički interfejsi su našli primenu u vremenski kritičnim sistemima (eng. *time-critical systems*) [21].

3. Korisnost i upotrebljivost

U ovom poglavlju će biti detaljnije obrađeni pojmovi korisnosti i upotrebljivosti. Kao što je već rečeno, korisnost se odnosi na skup funkcija koje sistem obezbeđuje, dok upotrebljivost govori koliko jednostavno se te funkcije mogu koristiti. Predstavljaju dve različite osobine interaktivnog sistema koje se ne mogu posmatrati odvojeno. Sistem se može vrednovati na više načina. Na primer, ako sistem ne implementira važne funkcionalnosti, tada je problem prvenstveno korisnost sistema. Ako je sistem kompletan u smislu da nudi širok spektar funkcija ali korisnici često greše u korišćenju tada je problem upotrebljivost. Sistem može biti lak za korišćenje (upotrebljiv) ali ne i koristan zbog toga što ima funkcije koje korisniku nisu potrebne. Obrnuto, sistem može imati sve potrebne funkcije (koristan) ali komplikovan za korišćenje (teže upotrebljiv).

Jednom kada se utvrdi da postoji problem upotrebljivosti korisničkog interfejsa, potrebno je dodatno konkretizovati jedan ili više problema i odrediti u kojoj meri oni otežavaju korišćenje interfejsa. Zbog toga su ustanovljeni atributi upotrebljivosti kako bi osobina bila merljiva i omogućila nalaženje pogodnih rešenja za specifične probleme.

Na ovaj način, upotrebljivost se može preciznije odrediti i izmeriti. Na primer, interfejs A može biti lakši za učenje, ili efikasniji, ili sigurniji za korišćenje od interfejsa B. Atributi upotrebljivosti nisu podjednako relevantni za različite grupe korisnika i različite tipove aplikacija. Zbog toga je važno razumeti potrebe i karakteristike ciljne grupe korisnika kako bi znali na koje osobine upotrebljivosti se treba fokusirati. Na primer, Veb sajt za trgovinu može biti namenjen širem krugu korisnika različitih karakteristika i tada je ključno da je što je više moguće lak za učenje. Program za bankarsko poslovanje koji koriste eksperti iz ekonomije ima ključni zahtev da je efikasan za korišćenje.

U opštem slučaju, važnost atributa upotrebljivosti je određena prirodom zadatka koji je potrebno obaviti korišćenjem interfejsa i karakteristikama ciljne grupe korisnika. Novi korisnici zahtevaju pre svega lakoću učenja interfejsa. Korisnici sa iskustvom zahtevaju prvenstveno efikasnost korišćenja. Upotrebljivost se može posmatrati u širem skupu karakteristika računarskog sistema kao što su funkcionalnost, bezbednost, performanse, cena, održivost, veličina i druge. Kod dizajna korisničkog interfejsa akcenat je na upotrebljivosti.

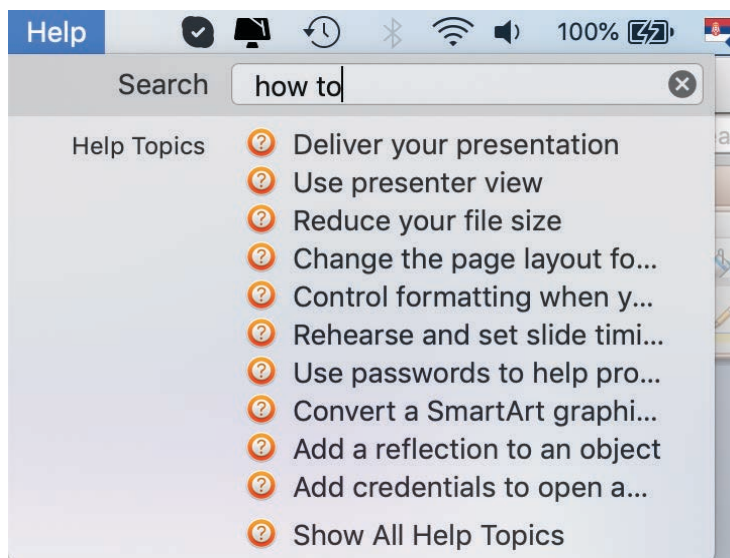
3.1. Lakoća učenja

Osobina upotrebljivosti koja govori koliko lako korisnici mogu koristiti sistem kada ga koriste prvi put, odnosno koliko lako se uči kako da se koristi sistem.

Pojava računara i programa namenjenih širem skupu korisnika koji nisu eksperti uslovlila je nastanak problema učenja korišćenja [7]. Praksa korišćenja korisničkog interfejsa je pokazala da, izuzev složenih, specijalizovanih programa, korisnici generalno ne preferiraju korišćenje uputstava, pomoći ili druge vrste podrške u učenju korišćenja korisničkog interfejsa. Kada prilaze interfejsu, korisnici generalno polaze od cilja koji treba da postignu. Na primer, pronalazak artikla koji žele da kupe u online prodavnici kao što je Amazon. U skladu sa ciljem, korisnik istražuje interfejs kako bi pronašao sredstva koja vode njegovom ispunjenju. Na primer, hijerarhijska i višekriterijumska pretraga kako bi se željeni proizvod pronašao, a zatim pregled samog artikla, uključujući njegove karakteristike i recenzije prethodnih kupaca. Ovakav način učenja je poznat pod nazivom učenje pokušajima (eng. *learning by doing*).

Ova vrsta učenja pretpostavlja da korisnik ima unapred definisan cilj i da je korišćenje interfejsa vođeno ostvarenjem cilja, a ne učenjem interfejsa kao takvog. Na ovaj način, učenje zapravo predstavlja posledicu pokušaja ostvarenja cilja korisnika. Princip može služiti kao smernica za razmišljanje pri dizajnu korisničkog interfejsa - interfejs svojim izgledom treba da pomogne korisnicima da izvršavaju ciljeve i neposredno govori korisnicima kako da ga koriste.

Ako korisnici ne nađu odgovarajuće opcije na interfejsu pokušajima, oni će potražiti pomoć. To znači da se i dizajnu pomoći korišćenja interfejsa mora posvetiti posebna pažnja. Korisnici generalno koriste pomoć sa konkretnim ciljem, problemom koji je nastao usled neispunjenja cilja i očekivanja konkretnog rešenja za dati problem. Zbog toga je pomoć u savremenim korisničkim interfejsima uglavnom realizovana kao sistem pitanja i odgovara sa mogućnošću pretrage po ključnim rečima.



Slika 21. Pomoć organizovana kao sistem pitanja i odgovora sa funkcijom pretrage gde su rezultati rangirani po učestalosti pitanja korisnika.

Način učenja korišćenja korisničkog interfejsa i traženja pomoći ima važne posledice na dizajn. Pre početka dizajna je potrebno pažljivo analizirati i definisati prave ciljeve korisnika. Ovde je ključan proces prikupljanja korisničkih zahteva koji će biti opisan u jednom od narednih poglavlja. Ukoliko su ciljevi korisnika pogrešno definisani pre početka dizajna, oni će imati probleme da nauče korišćenje interfejsa zbog toga što on ne ispujava ono što korisnici očekuju. Jednom kada su pravi ciljevi definisani, interfejs svojim izgledom mora u što je moguće većoj meri govoriti korisnicima kako da ga koriste. Ciljevi korisnika i akcije interfejsa koje ih realizuju se moraju podudarati. Primena principa dizajna (Poglavlje 1.4) doprinosi lakšem učenju interfejsa. Konačno, kada korisnici imaju problema u učenju kako da realizuju specifične ciljeve korišćenjem interfejsa, pomoć treba biti orijentisana ka cilju tako da pruža konkretne odgovore na probleme i sa mogućnošću pretrage.

Pored učenja pokušajima, korišćenje interfejsa se uči i učenjem posmatranjem (eng. *learning by watching*). Ovde se misli na posmatranje načina na koje drugi korisnici koriste interfejs. Sa jedne strane, većina korisničkih interfejsa je namenjena jednom korisniku u datom trenutku. Sa druge strane, postoje određene situacije u kojima je moguće posmatrati druge korisnike kako koriste određeni interfejs i na taj način učiti radnje na interfejsu. Primeri mogu biti standardne prečice na tastaturi (kombinacija *Alt* i *Tab* dugmadi za kretanje

između otvorenih prozora) ili specifični pokreti na dodirnom displeju (skaliranje).

3.1.1. Prepoznavanje i pamćenje

Predstavljaju dva značajna koncepta koja direktno utiču na lakoću učenja korišćenja interfejsa. Odnose se na ljudsku sposobnost identifikacije objekata ili akcija kroz korišćenje specifičnih elemenata koji mogu olakšati ili otežati identifikaciju elemenata interfejsa. Ovde je potrebno razlikovati identifikaciju elemenata interfejsa od lakoće pamćenja korišćenja interfejsa kao atributa upotrebljivosti. Koncepti zapravo predstavljaju rezultat načina rada memorije čoveka. Konkretno, objekat se lakše prepoznaje ukoliko postoji “vidljiv” trag o njemu.

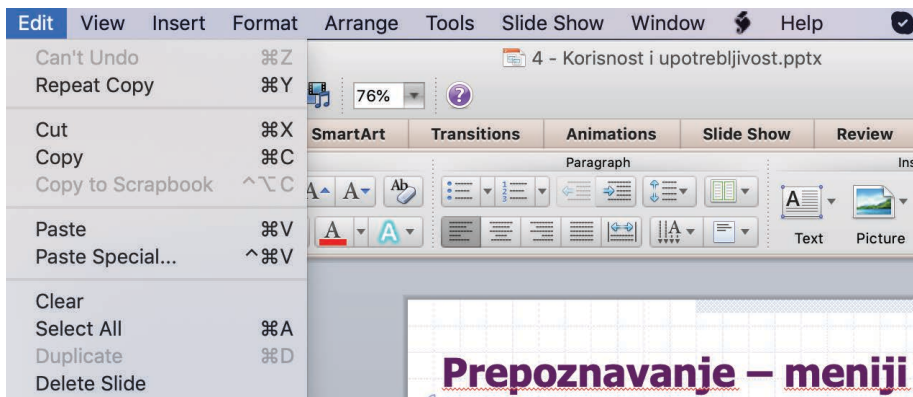
Prepoznavanje (eng. *recognition*) je identifikacija objekta na osnovu vidljivog traga ili karakteristike objekta. Pamćenje (eng. *recall*) je identifikacija objekta bez prisustva vidljivog traga ili karakteristike objekta, isključivo korišćenjem informacija iz ljudske memorije. Posledica važna za dizajn interfejsa je da je prepoznavanje znatno lakše od pamćenja za korisnika. Međutim, ovo ne mora važiti za specifične kategorije korisnika. Pre svega za korisnike-eksperte u određenoj oblasti.

Primer pamćenja je interfejs komandne linije (eng. *command line interface, CLI*). Ovi interfejsi su korišćeni u prvim računarskim sistemima. Korisnik unosi komande u veštačkom jeziku. Komande se moraju znati napamet. Drugim rečima, znanje o korišćenju se nalazi isključivo u memoriji korisnika i ne postoje vidljivih tragovi naredbi (Slika 22). Ovakvi interfejsi su generalno teži za učenje zbog toga što zahtevaju veće mentalno naprezanje kako bi se informacije dobavile i reprodukovale iz dugoročne memorije čoveka. Izuzetak mogu biti korisnici sa specifičnim znanjem i veštinama kao što su programeri gde interfejs komandne linije čini njihov rad efektivnijim i efikasnijim.

fx	=COUNTIF(D2:D220, TRUE)			
	A	B	C	D
215	Ariana	https://hiariana.c	web	TRUE
216	Wysa	www.wysa.io/	google	TRUE
217	Talkspace	talkspace.com/	google	TRUE
218	Scout	https://intermour	web	TRUE
219	K health	https://www.khea	web	TRUE
220	Clara (CDC)	https://www.cdc.	web	TRUE
221				152

Slika 22. Primer pamćenja - Excel operatori za matematičke proračune zahtevaju učenje specifične sintakse, dok sa druge strane ubrzavaju rad korisnika.

Primer prepoznavanja jesu vizuelni elementi korisničkih interfejsa. Vidljivi tragovi koji govore korisniku šta element radi su tekstualni nazivi akcija (meniji, dugmad i linkovi), simboli (ikone) ili kombinacija teksta i simbola. Zbog toga se kaže da je znanje o korišćenju ugrađeno u interfejs kako bi korisniku olakšalo identifikaciju objekata i akcija. Prepoznavanje čini interfejs lakšim za učenje zbog toga što vidljivi tragovi (asocijacije) pospešuju dobavljanje informacija iz dugoročne memorije čoveka.



Slika 23. Primer prepoznavanja - Meni kao element korisničkog interfejsa sadrži stavke opisane tekstualnim nazivima akcija (labelama). Labela predstavlja vizuelni trag koji opisuje akciju koju stavka pokreće.

3.1.2. Konceptualni modeli

U prethodnom poglavlju je uveden i definisan pojam konceptualnog modela. Kao takav on ispoljava značajan uticaj na lakoću učenja te će stoga biti detaljnije analiziran. Učenje korišćenja novog sistema za posledicu ima izgradnju mentalnog modela o tome kako sistem funkcioniše. Model sistema je predstava o tome kako sistem radi. Model definiše delove sistema i njihovu povezanost tako da zajedno realizuju funkcije sistema.

Na primer, model društvene mreže Tviter opisuje korisnike sistema, gde svaki korisnik ima spisak korisnika koje prati i spisak korisnika koji prate datog korisnika. Korisnik kreira tvitove koje vide njegovi pratioci i vidi tvitove koje kreiraju korisnici koje on prati. Korisnici i tvitovi imaju specifičnu strukturu i sadržaj. Dalje, postoje dozvoljene akcije nad tvitovima i korisnicima kao što su kreiranje, brisanje, ritvitovanje, citiranje i slično.

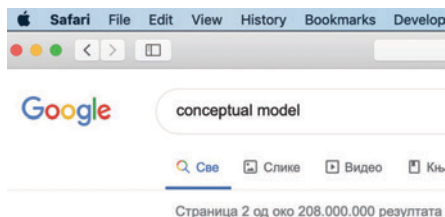
Dizajn korisničkog interfejsa se može posmatrati kroz relacije između:

- Modela sistema (eng. *system model*) koji opisuje implementaciju pozadinske logike sistema, odnosno kako sistem zapravo funkcioniše.
- Model interfejsa (eng. *interface model*) je posrednički model preko kojeg krajnji korisnici vide i koriste sistem.
- Model korisnika (eng. *user model*) opisuje način na koji korisnik vidi sistem, odnosno kako očekuje da sistem funkcioniše.

Konceptualno, ovde se može napraviti paralela sa softverskim inženjerstvom - SI (eng. *software engineering*) gde postoji razlika između interfejsa i implementacije modula. Dok interfejs nudi korisnicima specifične operacije nad modulom, implementacija nije vidljiva korisnicima i može biti raznovrsna.

Dok je u SI svrha razdvajanja interfejsa i implementacije smanjene kompleksnosti razvoja i održavanja softvera, u dizajnu interfejsa svrha je suprotna - model interfejsa treba biti što jednostavniji i što približniji modelu korisnika. Što je veća sličnost modela korisnika i interfejsa, interfejs će biti lakši za učenje.

Primer relacija između modela mogu biti Nazad (eng. *Back*) i Napred (eng. *Forward*) dugmad Veb pretraživača, odnosno korišćeni simboli strelica. Model sistema definiše Nazad kao povratak na stranicu koju je korisnik posetio prethodno iz istorijata posećenih stranica. Međutim, korišćenje identičnih simbola u okviru Veb sajta sa stranicama uređenim sekvencijalno, na primer skup stranica rezultata pretrage, može izazvati zabunu u modelu korisnika (Slika 24). Razlog tome je što identičan simbol ima različita značenje u dva konteksta aplikacija - prethodno posećena stranica kod Veb pretraživača (prvi slučaj) i prethodna stranica u sekvenci rezultata pretrage (drugi slučaj).



(a)



(b)

Slika 24. Modeli interfejsa i korisnika - Dva različita interfejsa koriste identičan simbol (horizontalne strelice) za dve različite operacije: (a) kretanje kroz istorijat posećenih stranica i (b) kretanje kroz skup rezultata pretrage rangiranih po relevantnosti.

3.1.3. Smernice dizajna za lakoću učenja

Na osnovu analize lakoće učenja kao odlike upotrebljivosti mogu se definisani odgovarajuće preporuke za dizajn korisničkog interfejsa:

- Ustanoviti prave cijeve korisnika pre početka dizajna.
- Interfejs treba da govori sam za sebe (vidljivost, priuštivost, povratna informacija).
- Dizajnirati za prepoznavanje pre pamćenja.
- Model interfejsa treba biti što približniji mentalnom modelu korisnika:
 - U skladu sa prethodnim iskustvom korišćenja sličnih interfejsa,
 - Koristeći pogodne metafore iz realnog sveta.
- Pomoć mora biti orijentisana ka cilju i omogućiti pretragu.

3.2. Efikasnost

Jednom kada korisnik nauči da koristi sistem, važno je da efikasno može obavljati zadatke korišćenjem sistema. Ovde se pod efikasnošću podrazumeva utrošak resursa za izvršenje zadataka na interfejsu, pre svega vreme potrebno za njihovo obavljanje. Kako bi lakše i brže procesirao informacije, čovek prirodno formira elementarne jedinice memorije ili informacija (eng. *chunk*) u radnoj memoriji (eng. *working memory*). Jedinice informacije su same po sebi dinamički i fleksibilni elementi i zavise od načina na koji su informacije prezentovane korisniku i prethodnog znanja. Kreirane jedinice informacija se zatim grupišu u krupnije, logičke celine. Kapacitet radne memorije pojedinca varira. Novija istraživanja pokazuju da taj kapacitet iznosi približno 4 ± 1 jedinica informacije [22]. Pored toga, podaci u ovoj memoriji se zadržavaju kratko vreme, od nekoliko sekundi do par desetina sekundi.

Sposobnost čoveka da formira jedinice informacija zavisi od načina na koji je ista prezentovana. Na ovaj način, dizajnom korisničkog interfejsa korisniku se može pomoći da lakše grupiše informacije u logičke celine što ima za posledicu efikasnije korišćenje interfejsa. Konkretno, organizacija informacija sadržanih u interfejsu u logičke i odvojene celine može ubrzati njegovo korišćenje.

Na primer, i kontinualna i grupisana sekvenca znakova ili cifara se opaža kao jedinica informacije. Razlika je u tome što prva grupa iziskuje veći mentalni napor da bi se razumela i zapamtila, dok je u kasnijoj grupi naprezanje manje i pamćenje olakšano (Slika 25). Slično tome, ako su grupe organizovane tako da se oslanjaju na postojeće znanje korisnika (kao što su skraćenice poznatih brendova) one će se pamtiti još lakše. Iz ovog razloga se složenije informacije kao što su telefonski brojevi, brojevi kreditnih kartica ili brojevi indeksa studenata prikazuju u specifičnom formatu kako bi se brže i bolje pamtile.

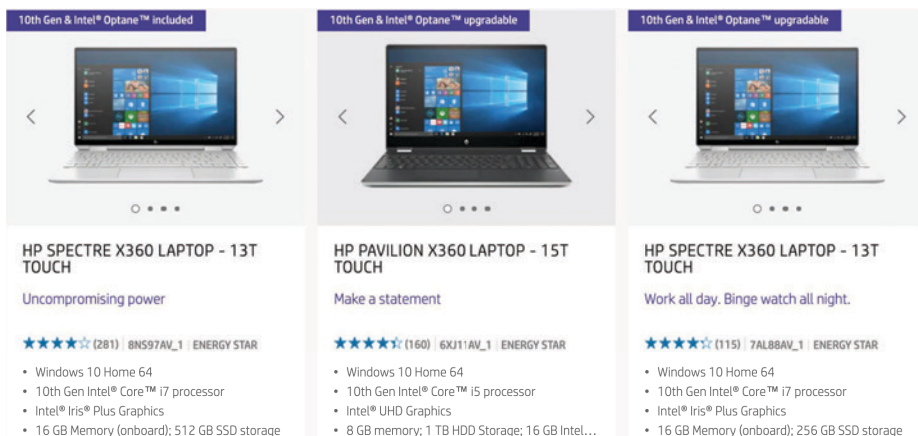
Teže za pamćenje: PGJAMRMBIWMBFDPCBB

Lakše za pamćenje: PGJ AMR MBI WMB FDP CBB

Najlakše za pamćenje: JPG RAM IBM BMW PDF BBC

Slika 25. Način grupisanja informacija u logičke jedinice koje čovek formira utiče na lakoću pamćenja informacija.

Jedinice informacija predstavljaju apstraktan pojam - interfejs svojim dizajnom, odnosno grupisanjem sadržaja može definisati ponavljajuće delove proizvoljne složenosti koji se opažaju kao jedinice (Slika 26).



Slika 26. Jedinica informacije predstavlja opis pojedinačnog proizvoda koji kombinuje vizuelni sadržaj (fotografija i simboli) sa tekstualnim opisom karakteristika.

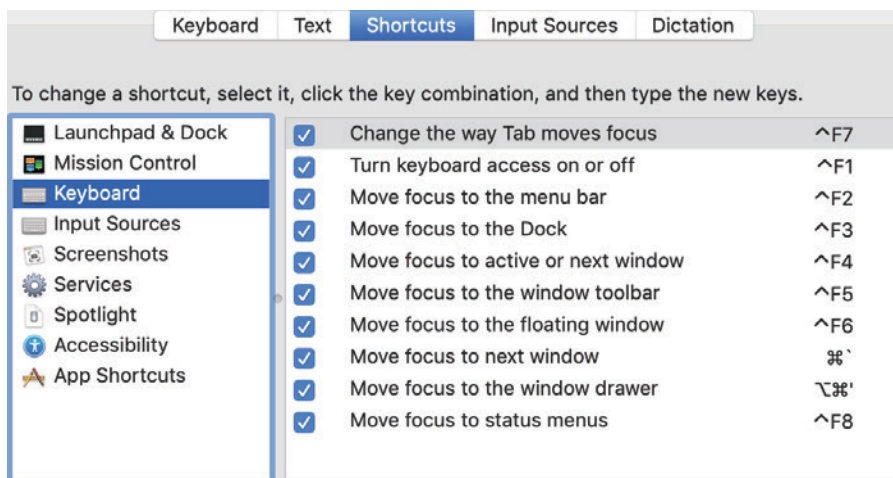
3.2.1. Prečice i podrazumevane vrednosti

Jedan od mehanizama za unapređenje efikasnosti korisničkog interfejsa jesu prečice (eng. *shortcuts*). Prečice su kombinacije tastera tastature ili pokreta koje pokreću akcije interfejsa i koje se lako pamte. Mogu se svrstati u dve grupe - prečice tastature i prečice pokreta.

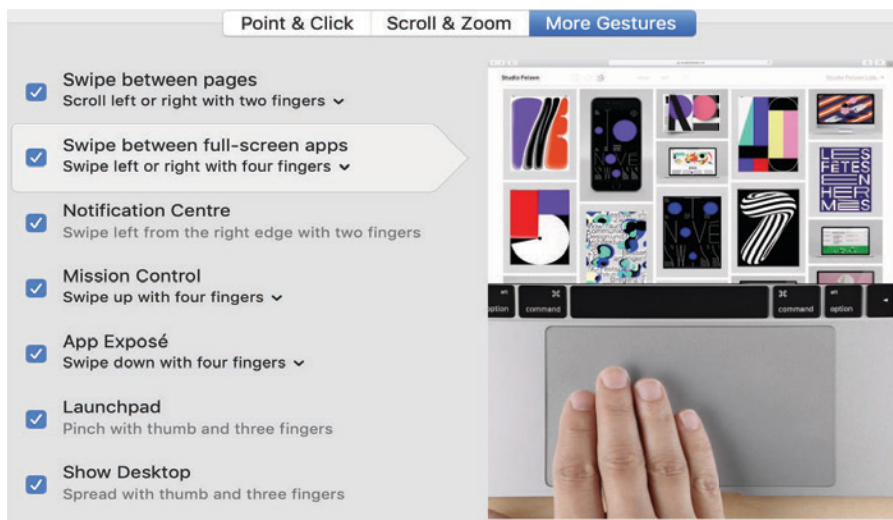
Prečice tastature su standardan način za ubrzavanje rada korisnika računara (Slika 27a). Postoje usvojene kombinacije tastera za često korišćenje operacije poput otvaranja objekta aplikacije (*open* kao Ctrl + O), kreiranja objekta (*new* kao Ctrl + N) ili čuvanje izmena nad objektom (*save* kao Ctrl + S). Prilikom kreiranja prečica treba voditi računa o složenosti kombinacija. U opštem slučaju, povećanje broja tastera koji čine kombinaciju otežava učenje interfejsa.

Prečice pokreta su kombinacije prstiju ruke i njihovih specifičnih pokreta za obavljanje specifičnih radnji nad interfejsom (Slika 27b). Pojavile su se sa unapređenim karakteristikama dodirnih uređaja. U zavisnosti od uređaja, koriste se kod dodirnih displeja telefona i tableta, i kod dodirnih panela računara. Slično prečicama tastature, usloznjavanje strukture pokreta prečice može uzrokovati poteškoće u učenju korišćenja interfejsa.

Uzimajući u obzir ograničenja koja mogu nastati primenom prečica, savremene aplikacije pružaju korisnicima pomoć i fleksibilnost u njihovom korišćenju (Slika 27).



(a)

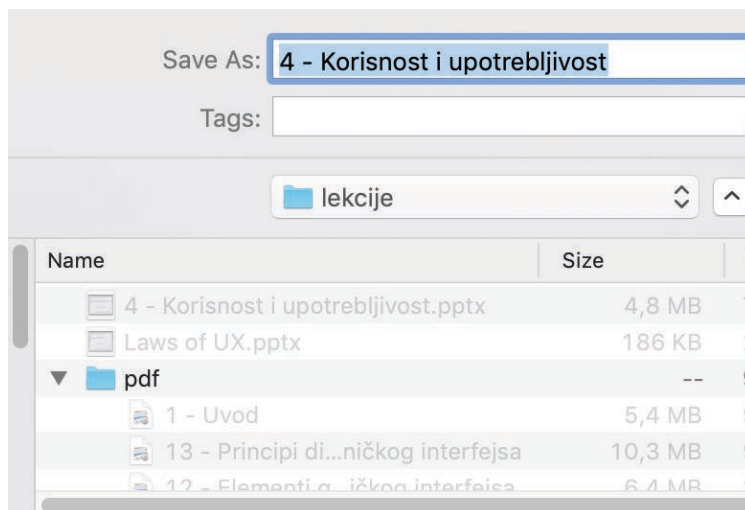


(b)

Slika 27. Primer prečica (a) tastature i (b) pokreta.

Podrazumevane vrednosti su one vrednosti koje korisnici najčešće navode prikom unosa podataka aplikacije, najčešće u formama za unos. Ove vrednosti se dobijaju analizom istorijata unosa podataka korisnika na specifičnim mestima u aplikaciji. Mogu doprineti efikasnosti na različite načine. Podjednako ubrzavaju rad korisnika sa i bez iskustva. Zatim mogu redukovati greške i ponavljanje unosa tako što daju primere ispravnog unosa podataka. Poželjno je da podrazumevane vrednosti budu selektovane onog trenutka kada

korisnik unosi podatke (Slika 28). Na taj način korisnici mogu brže uneti novu, željenu vrednost, drugačiju od ponuđene.

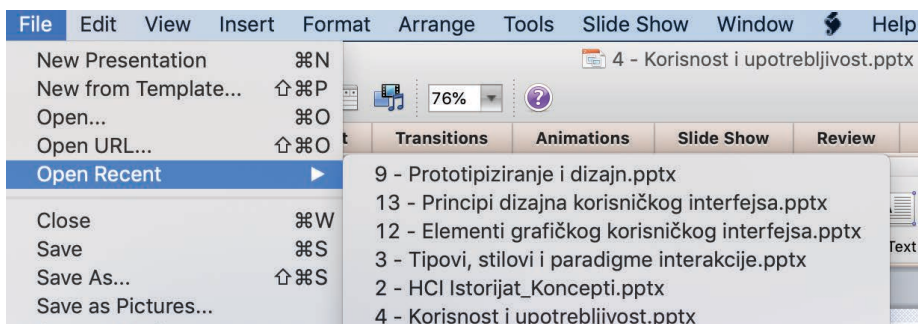


Slika 28. Podrazumevane vrednosti su inicijalno selektovane kako bi dodatno ubrzale korišćenje interfejsa.

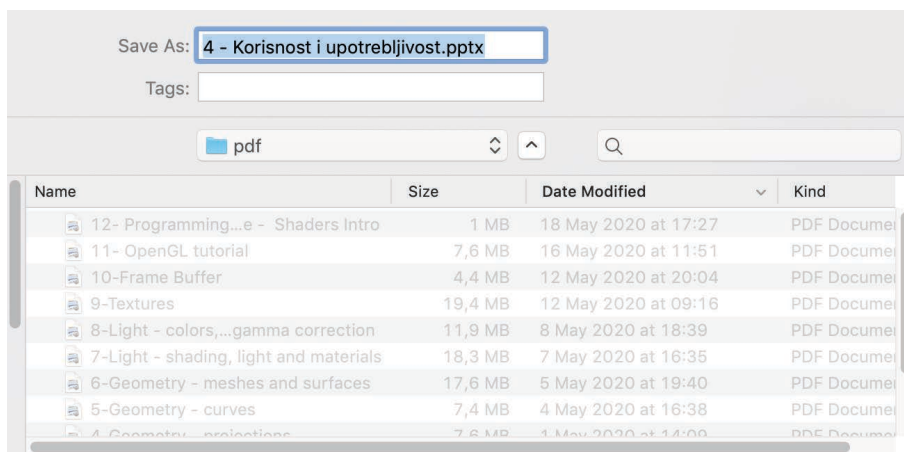
3.2.2. Istorijat i agregacija

Unos podataka korisnika je zasnovan na principu vremenske lokalnosti [7]. Princip kaže da će se korisnik vrlo verovatno na konkretnom mestu aplikacije uneti neku od vrednosti koju je prethodno uneo. Podjednako se odnosi na korišćenje komandi, datoteka i drugih objekata interfejsa. Zbog toga sistem čuva istorijat rada korisnika kako bi ponudio skoro korišćene objekte i time ubrzao korišćenje interfejsa tako što korisnik ne bi tražio ove objekte (Slika 29). Opcije su rangirane po opadajućem redosledu na osnovu vremena korišćenja u prošlosti (skorije opcije su pri vrhu).

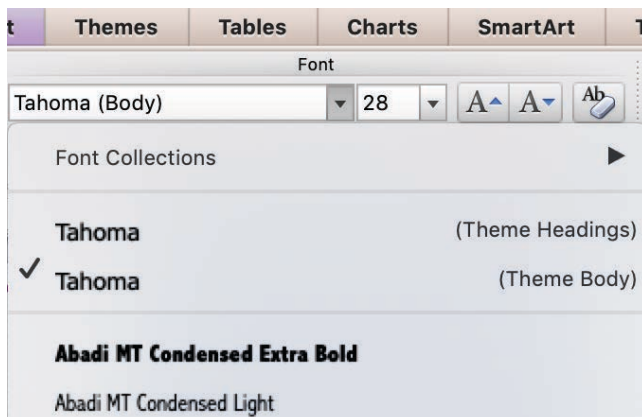
Ovaj princip ima šire značenje u sistemima preporuke (eng. *recommender systems*) koji polaze od pretpostavke da će se korisnik u budućnosti ponašati slično kao u prošlosti. Tako ovi sistemi analiziraju istorijat aktivnosti korisnika i preporučuju objekte ili aktivnosti slične onima koji su već korišćeni.



(a)



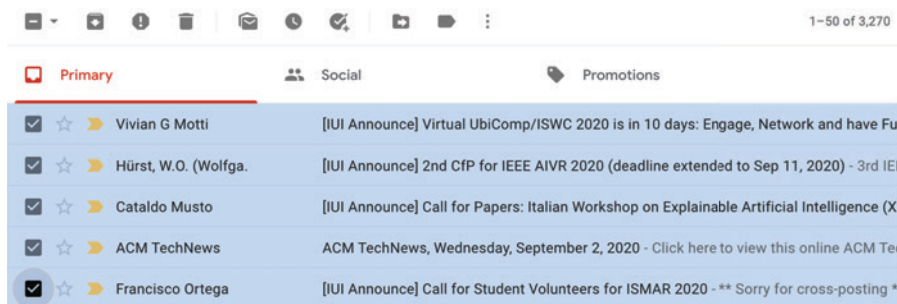
(b)



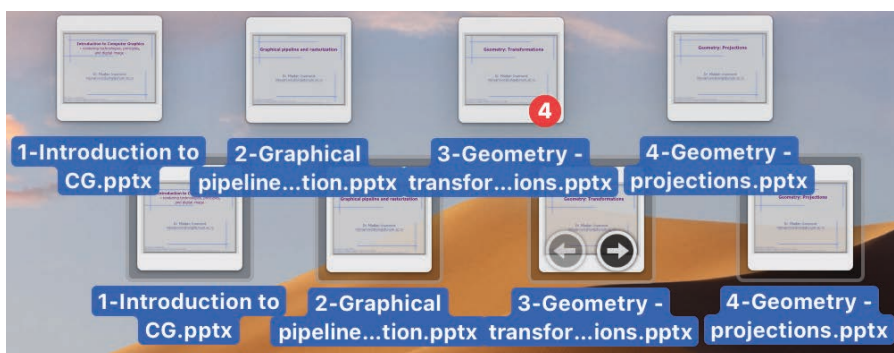
(c)

Slika 29. Istorijat korišćenih objekata kao što su (a) dokumenti, (b) direktorijumi, ili (c) fontovi ubrzava korišćenje interfejsa.

Agregacija podrazumeva izbor i manipulisanje većim brojem objekata u isto vreme kao grupe. Objekat može biti predmet akcije na interfejsu, kao što je datoteka, objekat interfejsa, ili komanda. Princip se svodi na višestruku selekciju i upravljanje većim brojem umesto pojedinačnim objektom interfejsa. Na ovaj način se ostvaruje ušteda u vremenu i ubrzava korišćenje interfejsa (Slika 30). Sa druge strane, prethodno je potrebno sagledati da li objekti po svojoj prirodi omogućavaju primenu agregacije.



(a)



(b)

Slika 30. Primer agregacije - (a) upravljanje porukama elektronske pošte i (b) prevlačenje grupe objekata korišćenjem miša.

3.2.3. Smernice dizajna za efikasnost

Na osnovu analize efikasnosti kao atributa upotrebljivosti mogu se definisani odgovarajuće preporuke za dizajn korisničkog interfejsa:

- Organizacija sadržaja u logičke celine jedinica informacija.
- Organizacija teksta:
 - Kraći i logički paragrafi odvojeni praznim prostorom,

- Zaglavlja koja se ističu u odnosu na ostatak tekst (font, veličina, boja),
- Naglašene ključne reči (podebljano - *bold*, kurziv - *italic*),
- Neuređene ili numerisane liste za elemente nabiranja,
- Paragraf-sažetak ispred dužih sekcija teksta,
- Kraće linije teksta (približno 50-75 karaktera),
- Grupisanje slova ili brojeva kao što su lozinke, ključevi dozvola, brojevi kreditnih kartica, bankovni računi, brojevi telefona i datumi.
- Organizacija multimedijalnog sadržaja:
 - Multimedijalni sadržaj (tekst, grafika, video, dugmad, i drugi),
 - Grupisanje po različitim vizuelnim kriterijumima kao što su sadržaj, blizina, oblik, veličina, simetrija.
- Prečice tastature i pokreta.
- Podrazumevane vrednosti.
- Istorijat korišćenja objekata aplikacije.
- Agregacija.

3.3. Greške

Prilikom korišćenja interfejsa očekivano je da korisnici prave greške. Pojava grešaka utiče na upotrebljivost korisničkog interfejsa. Iz tog razloga jedan od ciljeva dizajna interfejsa jeste smanjenje broja grešaka koje korisnik može napraviti. Kako bi se pravilno pristupilo rešavanju ovog problema potrebno je odrediti: a) vrste grešaka koje mogu nastati, b) koliko su nastale greške ozbiljne i u kojoj meri utiču na normalan rad korisnika, i c) koje su posledice nastanka grešaka po sistem i da li se sistem može dalje koristiti.

3.3.1. Vrste grešaka

Prema uzroku nastanka, greške se mogu podeliti u dve grupe [23]:

- Greške u izvršavanju naučenih procedura, koje mogu biti:
 - Izvršne greške (eng. *slips*),
 - Greške memorije (eng. *lapse*).
- Greške u izvršavanju novih procedura, vezane za planiranje i zaključivanje (eng. *mistake*).

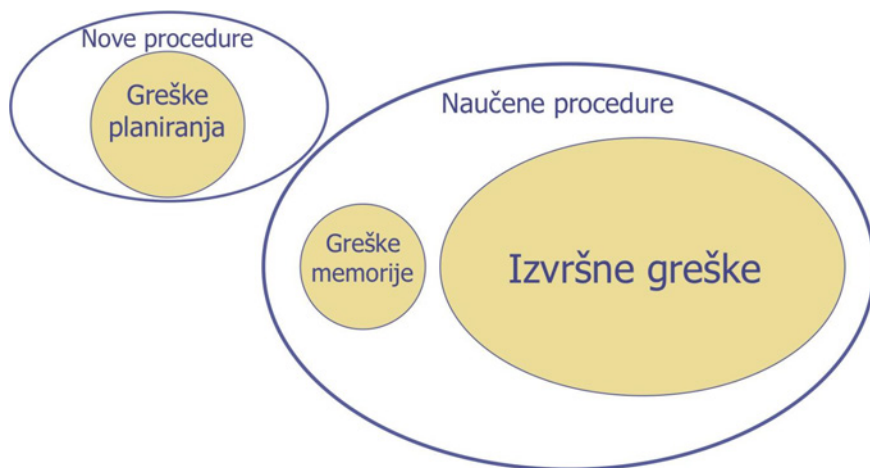
Izvršne greške i *greške memorije* nastaju u toku izvršavanja akcija na interfejsu koje su korisniku poznate, odnosno prethodno naučene. Na primer, prevlačenje pogrešne datoteke i crtanje pogrešnog objekta su greške kod kojih je korisnik prethodno usvojio operacije prevlačenja objekata ili crtanja oblika kao takve. Dva tipa grešaka se razlikuju na osnovu izvora greške.

Kod izvršnih grešaka uzrok je devijacija u kontroli ili izvršavanju procedure. Na primer, korisnik bira jednu akciju umesto druge u okviru zadatka. Obično nastaju kada korisnici ne posvećuju dovoljno pažnje zadatku koji trenutno obavljaju. Primeri mogu biti unos pogrešnog slova, broja ili reči, odabir pogrešne opcije kod selekcije, pritiskanje pogrešnog tastera miša ili zadavanje pogrešne reči za pretragu.

Uzrok grešaka memorije je zaboravljanje, kao što je zaboravljanje cilja tekuće akcije ili zaboravljanje pozicije u kojoj se korisnik nalazi u toku izvršenja složenije procedure.

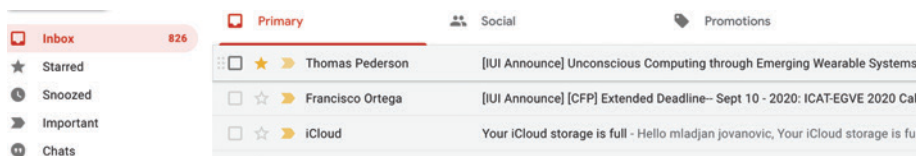
Greške u izvršavanju novih procedura nastaju kao posledica pogrešnog planiranja ili donošenja odluka. Drugim rečima, primena pravila u situaciji u kojoj ono nije primenjivo ili pogrešno zaključivanje. Ovaj tip grešaka se tipično javlja prilikom korišćenja interfejsa prvi put, odnosno kod učenja korišćenja. Na primer, izbor pogrešne ikone ili dugmeta za obavljanje operacije koja ispunjava specifičan cilj korisnika.

Izvršne greške i greške memorije se javljaju češće nego greške planiranja i zaključivanja. Razlog tome je što većinu vremena provodimo izvršavajući naučene procedure u odnosu na učenje novih, kako u svakodnevnom životu tako i korišćenju tehnologije [23]. Statistika varira, u zavisnosti od vrste interfejsa i zadatka. U opštem slučaju, približno 60% grešaka su izvršne greške ili greške memorije [23].



Slika 31. Tipologija i zastupljenost grešaka. Veličina oblika elipse Ven dijagrama odgovara udelu odgovarajuće vrste grešaka. Preuzeto i adaptirano iz [23].

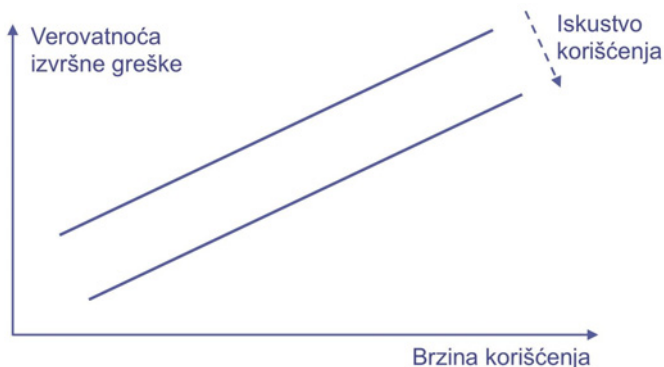
Dve najčešće vrste grešaka izvršavanja koje nastaju u korisničkim interfejsima su greška sličnosti akcija i greška višeznačne akcije. U prvom slučaju, dve akcije izgledaju slično i korisnik greškom pokreće jednu umesto druge. Primer mogu biti simboli ikona za upravljanje porukama elektronske pošte (Slika 32). U drugom slučaju, jedna akcija ima različita značenja u okviru jednog korisničkog interfejsa. Čest primer ovakve greške jeste korišćenje jednog simbola za dve ili više ikona koje pozivaju različite funkcije interfejsa.



Slika 32. Primer greške sličnosti akcija - tri ikone za označavanje poruke imaju različito značenje (eng. *check, start, important*), dele slične vizuelne karakteristike i raspoređene su blizu.

Glavni uzrok izvršnih grešaka jeste nedovoljna pažnja prilikom obavljanja naučene procedure ili ometanje pažnje korisnika. Izvršne greške se mogu opisati krivom koja pokazuje odnos brzine korišćenja (eng. *speed*) i tačnosti izvršenja zadatka (eng. *accuracy*) [23]. Što se brže donose odluke, veća je verovatnoća da će biti izabrana pogrešna opcija ili izvršena pogrešna radnja na interfejsu. Ako se više vremena iskoristi za donošenje odluke, manja je verovatnoća da će akcija biti izvršena pogrešno.

Ovaj primer ilustruje i međuzavisnost efikasnosti i grešaka kao atributa upotrebljivosti.

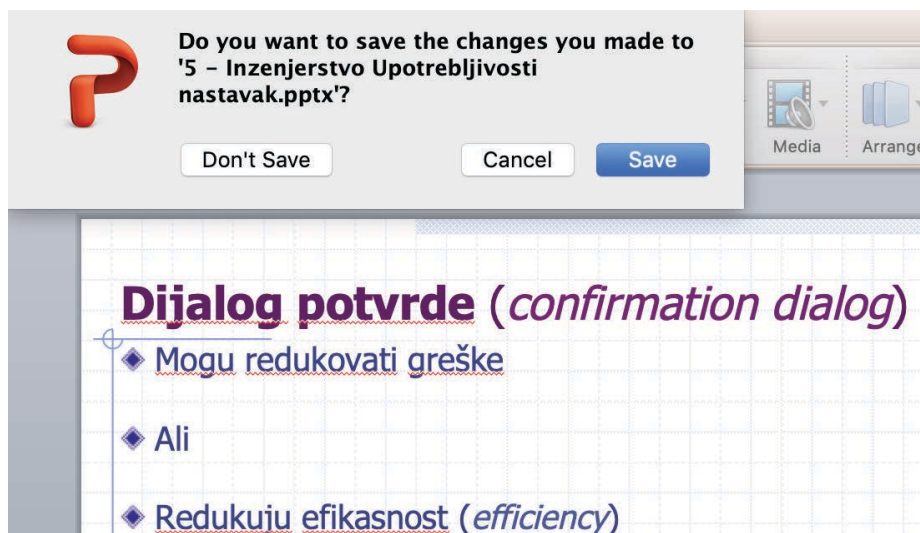


Slika 33. Balans brzine i kvaliteta korišćenja interfejsa je linearna funkcija. Brže korišćenje generalno povećava verovatnoću nastanka greške. Iskustvo čini brže korišćenje i preciznijim. Oblik krive zavisi i od konkretnog zadatka.

3.3.2. Prevencija grešaka

Analiza grešaka koje mogu nastati u toku korišćenja interfejsa je važna za definisanje preporuka ili prevencije grešaka u procesu kreiranja interfejsa. Pre svega je potrebno jasno razdvojiti akcije na interfejsu, odnosno izbegavati akcije sa sličnim opisima kako korisnik ne bi birao pogrešne opcije. Dakle, različite akcije na interfejsu moraju različito izgledati. Ovo se posebno odnosi na definisanje naziva dugmadi i izbor simbola za ikone. Pored toga, potrebno je razdvojiti kritične akcije čije pokretanje i izvršenje može imati ozbiljnije posledice po sistem (na primer brisanje) od često korišćenih akcija čije posledice po rad sistema ne moraju biti ozbiljne (na primer čuvanje promena).

Dijalog potvrde (eng. *confirmation dialog*) pre izvršenja operacije je često korišćen mehanizam prevencije grešaka. Sa jedne strane, dijalozi mogu smanjiti verovatnoću nastanka greške. Sa druge strane, mogu redukovati efikasnost, odnosno brzinu korišćenja interfejsa. Konkretno, veći broj akcija korisnika na interfejsu (klikova) povećava vreme korišćenja i naprezanje korisnika. Opšte pravilo je da se dijalozi potvrde koriste samo za kritične akcije koje mogu imati značajnije posledice po rad sistema (gubitak podataka kod čuvanja određenih promena ili brisanja) kao što prikazuje Slika 34. Za često korišćene komande koje nemaju ozbiljnije posledice po sistem ne treba koristiti dijalog potvrde zbog toga što može usporiti rad korisnika.

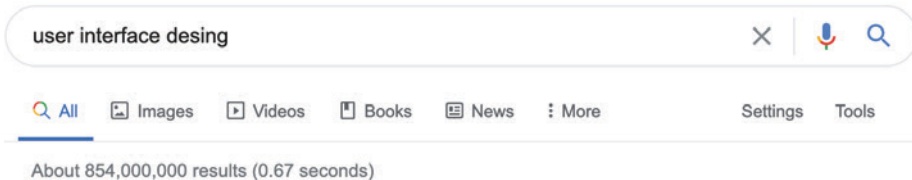


Slika 34. Dijalog potvrde koji podseća korisnika da sačuva promene pre zatvaranja dokumenta.

3.3.3. Poruke o grešci

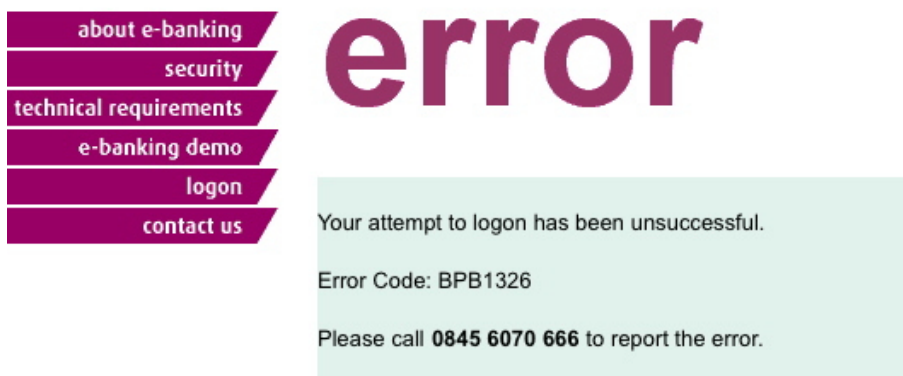
Greške prilikom korišćenja interfejsa se ne mogu eliminisati u potpunosti. Zbog toga je važno informisati korisnike na odgovarajući način kada greška nastane kako bi: a) identifikovali grešku, b) ustanovili razlog njenog nastanka, i c) bili informisani o načinu na koji je moguće izbeći grešku u daljem korišćenju interfejsa.

Poruka o grešci mora biti precizna i sažeta u opisu jedne konkretne greške koja je nastala. Pored toga, poruka se mora nalaziti blizu mesta na interfejsa gde je greška nastala. Često se dešava da korisnik ne razume zbog čega je greška nastala, te je preporuka da se u okviru poruke o grešci opiše akcija korisnika koja je izazvala grešku. Na ovaj način se korisniku pomaže da identifikuje razlog nastanka greške. Jezik poruke o grešci mora biti poznat i razumljiv za korisnika. Potrebno je izbegavati stručne termine i fraze koji mogu biti nepoznati i složeni za razumevanje krajnjim korisnicima. Osim što korisnika obaveštava o problemu koji je nastao, poruka mora biti konstruktivna i ponuditi jedno ili više rešenja za oporavak od greške. Na kraju, poruka mora biti učtiva prema korisniku, formulisana tako da ne ukazuje da je greška krivica korisnika (Slika 35).

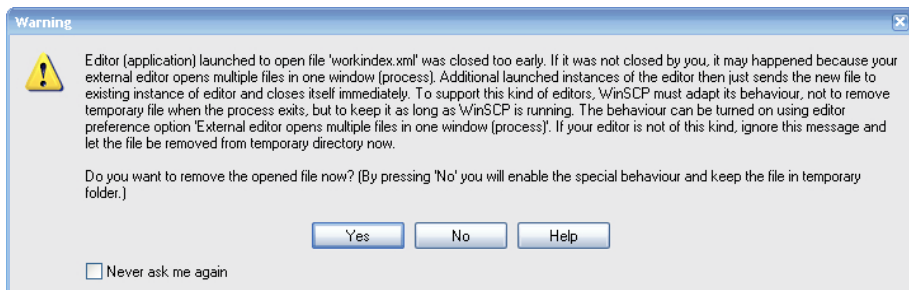


Showing results for **user interface design**
Search instead for **user interface desing**

(a)



(b)



(c)

Slika 35. Poruke o grešci: (a) poruka je konstruktivna tako što ukazuje korisniku na ispravan upit; (b) poruka sadrži istaknutu reč greška, implicira krivicu korisnika i sadrži tehničke termine (oznaka greške); i (c) poruka nije dovoljno precizna.

Izvor (b) i (c): <https://www.userfocus.co.uk/articles/errormessages.html>.

3.3.4. Smernice dizajna za sigurnost

Na osnovu analize grešaka kao faktora sigurnosti korisnika mogu se izdvojiti odgovarajuće preporuke za dizajn korisničkog interfejsa:

- Prevencija grešaka pre njihovog nastanka:
 - Različite akcije na interfejsu moraju različito izgledati,
 - Jedna akcija na interfejsu mora izgledati identično na različitim mestima,
 - Dijalog potvrde koristiti samo za kritične operacije interfejsa.
- Jezik poruke o grešci mora biti precizan, lokalni, razumljiv za korisnika, konstruktivan, učitiv i neutralan.

4. Osnove psihologije korisnika

U osnovi korišćenja tehnologije su različiti oblici kognitivnog naprezanja čoveka. Zbog toga je važno razumeti i poznavati karakteristike i ograničenja kognitivnog aparata korisnika. Ovo pre svega omogućava otkrivanje i objašnjenje problema u korišćenju interfejsa ili uređaja. Osim toga, poznavanje i razmatranje kognitivnih procesa vodi ka dizajnu upotrebljivijih i privlačnijih interaktivnih sistema. Kognitivni procesi se mogu posmatrati kao [12]:

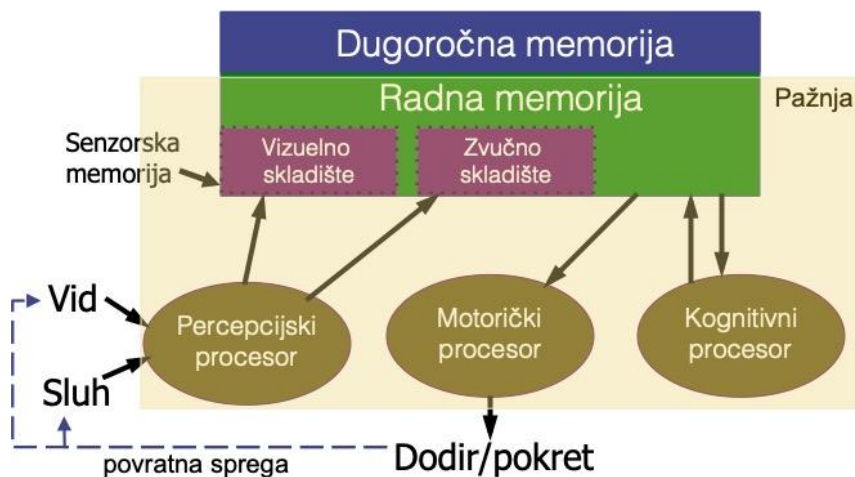
- pažnja,
- percepcija i prepoznavanje,
- memorija,
- čitanje, pisanje i slušanje,
- rešavanje problema, planiranje, rezonovanje, donošenje odluka, i učenje.

U ovom poglavlju, naglasak će biti na prva tri procesa zbog toga što imaju značajniji uticaj na korišćenje korisničkog interfejsa.

4.1. Model procesiranja čoveka

Kognitivni procesi su međuzavisni i veći broj se može odigravati u određenoj aktivnosti. Na primer, pretraga sadržaja u korisničkom interfejsu podrazumeva određeni nivo pažnje kako bi se primetili i prepoznali odgovarajući navigacioni elementi korisničkog interfejsa (ikone ili tekstualni linkovi) koji se vizuelno skeniraju i čitaju, zatim pamćenje trenutne pozicije u okviru hijerarhije veb sajta (orijentacija) i na kraju čitanje i rezonovanje o pronađenom sadržaju.

Model procesiranja čoveka je najraniji i još uvek široko prihvaćen metod da se šematski i pojednostavljeno opiše čovekov kognitivni aparat na način razumljiv i upotrebljiv inženjerima [24]. Opis naravno predstavlja apstrakciju složenog biološkog aparata, a ne konkretnu anatomiju.



Slika 36. Model procesiranja čoveka. Preuzeto i adaptirano iz [24].

Vizuelni i zvučni signali se detektuju putem čula vida i sluha i smeštaju u odvojene lokacije u memoriji senzora. Percepcijski procesor analizira prikupljene signale i pokušava da ih interpretira, odnosno prepozna simbole kao što su znakovi, reči, glasovi i ikone. Prepoznavanje je podržano i iz dugoročne memorije koja čuva naučene informacije. Kognitivni procesor dalje analizira prepoznate simbole, vrši poređenja i donosi odluke. Može i dobavljati i analizirati informacije iz radne memorije. Proces koji doživljavamo kao razmišljanje se najvećim delom odvija u kognitivnom procesoru. Motorički procesor dobija naredbe iz kognitivnog procesora i pokreće mišiće za izvršenje akcija (pokreti ruke, prstiju, glave i očiju). U tom trenutku se formira povratna sprega interakcije korisnika sa interfejsom - rezultat akcije (selekcija dugmeta ili pomeranje objekta) registruju čula, signali se analiziraju, donosi odluka i koriguje pokret (izbor sledećeg dugmeta ili naredni pomeraj objekta). Pažnja se može posmatrati kao pozadinski proces koji inicira, čini aktivnim i utiče na rad opisanih mehanizama.

Model kao celina komponenti može funkcionisati u dva režima rada - paralelno i redno procesiranje. Režim je određen tipom i složnošću aktivnosti koje čovek obavlja.

Paralelno procesiranje podrazumeva rad većeg broja procesora (Slika 36) u datom trenutku vremena. Primer aktivnosti može biti korišćenje simulatora vožnje ili letenja, ili aplikacija virtuelne stvarnosti. Pomenute aktivnosti istovremeno angažuju čula, procesore i aktuator korisnika. Ovaj tip procesiranja je karakterističan za aplikacije koje angažuju veći broj čula korisnika kako bi proizveli snažan korisnički doživljaj.

Redno procesiranje opisuje sekvencijalan rad procesora - jedan procesor je aktivan u datom trenutku. Karakteristično je za većinu korisničkih interfejsa nezavisno od domena aplikacije. Primeri su standardne operacije nad korisničkim initerfejsima poput pritiska tastera miša, selekcije dugmeta ili ikone, ili izbor stavke menija.

Radna ili kratkoročna memorija ima ograničen kapacitet i vreme trajanja informacija (reda veličine nekoliko sekundi do nekoliko desetina sekundi). Sa druge strane je odlikuje veća brzina pristupa (reda veličine milisekunde). Dugoročna memorija ima veći kapacitet, ali sporije vreme pristupa od radne memorije.

Dve memorije komuniciraju međusobno po principu diskriminacije (eng. *discrimination principle*). Dobavljanje informacija iz dugoročne memorije zavisi od prisustva tragova ili asocijacija. Drugim rečima, sadržaji radne memorije pokreću akcije na osnovu asocijativnih konekcija sa dugoročom memorijom. Prepoznavanje i pamćenje (Poglavlje 3.1.1) su zapravo posledica opisanog mehanizma. Kod pamćenja ne postoje vidljivi tragovi u radnoj memoriji i zbog toga se informacije reprodukuju iz dugoročne memorije. U prepoznavanju, tragovi o informacijama u radnoj memoriji pospešuju, odnosno olakšavaju njihovo dobavljanje iz dugoročne memorije. Trag predstavlja bilo koju informaciju povezanu sa naučenim objektom, akcijom ili procedurom kao što je simbol ikone, ili naziv dugmeta ili stavke menija.

4.2. Pažnja

Pažnju možemo definisati kao kognitivni proces izbora stvari na koje možemo da koncentrišemo u datom trenutku. Pažnja podrazumeva angažovanje većeg broja komponenti modela procesiranja u datom trenutku (Slika 36). Sa obzirom da angažuje druge kognitivne procese, pažnja je ograničen resurs čoveka koji je potrebno uzeti u obzir prilikom razvoja interaktivnih sistema, odnosno korisničkih interfejsa. Kao proces omogućava fokus na informaciju koja je relevantna za trenutnu aktivnost korisnika u skladu sa ciljem. Odavde proizilazi da na ovaj proces utiču dva glavna faktora:

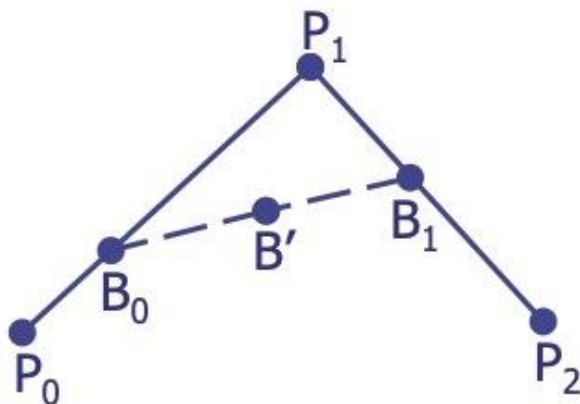
- precizno definisan cilj korisnika,
- jasno istaknuta informacija relevantna za cilj, odnosno aktivnost korisnika.

U skladu sa definisanim cijeovima, korisnik traži informacije na interfejsu za koje veruje da vode njihovom ispunjenju. Zbog ovoga je važno definisati prave ciljeve korisnika pre početka dizajna korisničkog interfejsa, a zatim pristupiti kreiranju servisa koji ispunjavaju definisane ciljeve. Ovde je ključno pravilno

prikupiti i formulisati korisničke zahteve. Postupci i tehnike će biti opisani u Poglavlju 6.

Način na koji je informacija vezana za ispunjenje cilja prezentovana utiče na optimalno korišćenje pažnje. Konkretno, u kojoj meri je jednostavnije ili komplikovanije pronaći pravu informaciju na interfejsu. Ovo se prvenstveno odnosi na zadatke pretrage informacija. Zbog toga je važno pravilno struktuirati i rasporediti informacije na korisničkom interfejsu. U slučaju GKI primenjuju se Geštalt principi vizuelne percepcije čoveka opisani u ovom poglavlju. Percepcija podrazumeva način na koji se informacije prikupljene od strane čula (vid, sluh, dodir) transformišu i interpretiraju kao doživljaji objekata, događaja, zvukova i ukusa [25]. Predstavlja složen proces, koji kombinuje memoriju, pažnju i jezik. Vid predstavlja najdominantnije čulo u pogledu količine prikupljenih informacija za osobe sa normalnim funkcionisanjem vida. Slede sluh i dodir.

Pažnja je specifičan kognitivni proces koji se može odvijati na dva načina - selektivna pažnja i podeljena pažnja. Selektivna pažnja se odnosi na opažanje specifičnog signala (stimulusa) u obilju postojećih. Podeljena pažnja podrazumeva opažanje većeg broja signala istovremeno u mnoštvu postojećih.

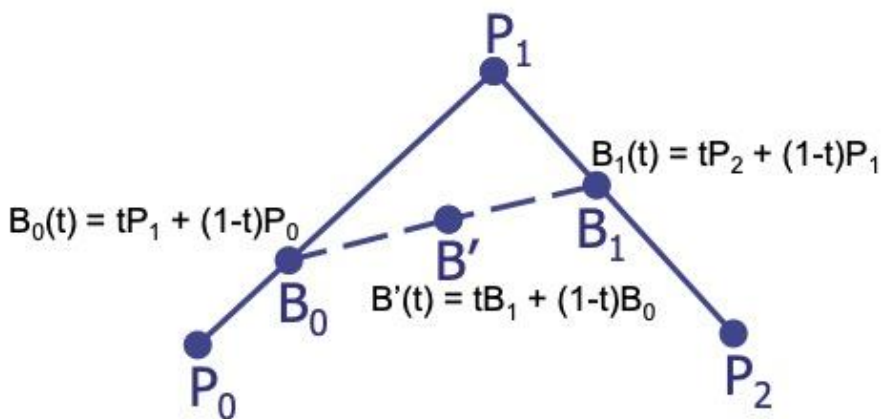


$$B_0(t) = tP_1 + (1-t)P_0$$

$$B_1(t) = tP_2 + (1-t)P_1$$

$$B'(t) = tB_1 + (1-t)B_0$$

(a)



(b)

Slika 37. Upravljanje pažnjom - (a) podeljena pažnja kod koje je fokus istovremeno i naizmenično na krivu i formule, i (b) približavanjem formula krivoj informacije se lakše i brže opažaju zbog toga što je fokus na jednoj celini (selektivna pažnja). Izvor: Computer Graphics course, Univerzitet Singidunum.

4.3. Percepcija

U osnovi, opažanje (percepcija) korisnika odvija se na tri nivoa - motoričko-instiktivnom, emotivnom i intelektualnom. Sa evolutivnog aspekta, motoričko-instiktivni centar za obradu impulsa dobijenih od čulnih organa je najstariji, odnosno najrazvijeniji. Ovo se jasno može videti po složenosti i broju paralelnih operacija koje ovaj centar istovremeno obavlja, kao i po brzini njegovih reakcija. To je i logično, s obzirom na to da preživljavanje živih bića i njihova interakcija sa okolinom prvenstveno zavise od ovog centra. Na primer, ukoliko čoveka iznenada napadne divlja životinja, ili se jednostavno spotakne pri hodu, motorički centar je odgovoran za brzu reakciju zaštite glave i vrata, odnosno amortizovanja pada. Naziv za perceptivni utisak ovog centra je oset ili osećaj.

Emotivni ili osećajni centar čoveka, kao evolutivno znatno mlađi, ujedno je i znatno manje razvijen, kao i znatno sporiji od motoričkog centra. Zadatak ovog nivoa opažanja je, prvenstveno, da na osnovu podataka dobijenih od motoričko-instiktivnog centra izvrši klasifikaciju višeg nivoa. Ova klasifikacija je uglavnom dvosmerna, odnosno određuje da li je nešto prijatno ili ne. Za razliku od motoričko-instiktivnog centra, emotivni centar ne poseduju sve životinje, već samo one razvijenije. Naziv za perceptivni utisak na emotivnom nivou je osećanje.

Konačno, intelektualni centar čoveka, iako posebno cenjen i uvažavan, evolutivno je najmlađi, odnosno najmanje je razvijen, a samim tim je i najsporiji. Za razliku od prethodno navedenih centara, intelektualni centar nema mogućnost paralelne obrade više stimulusa, već se njegov rad svodi na upoređivanje dva objekta. Ovakva jednostavnost je ujedno bila i osnova za razvoj računarstva, a kasnije i veštačke inteligencije. Postojanje ovog centra je karakteristično za čoveka kao vrstu.

Razumevanje navedene strukture, odnosno hijerarhije čovekovog opažanja kritično je za pravilno razumevanje i organizaciju interakcije. Motorički centar je neuporedivo brži i moćniji od intelektualnog, ali i srazmerno sporiji u prihvatanju novih znanja. Na primer, za početnika u procesu obuke za vožnju automobila gotovo je nemoguće da istovremeno gleda kroz vetrobran, kontroliše volan, menjač, papučice gasa/kočnice/kvačila, ručice za pokazivače pravca i nadgleda retrovizore. Intelektualni centar može svaku od navedenih akcija usvojiti za svega nekoliko minuta, ali njegovi kapacitet i struktura nisu dovoljni za paraleleno izvršavanje svih navedenih operacija. Međutim, kada se ova znanja premeste iz intelektualnog u motorički centar, čovek je u stanju da bez opterećenja obavlja te operacije, kao i da paraleleno razmišlja o stvarima potpuno nevezanim za vožnju. Kada se ovaj princip razume, onda i ne čudi

zbog čega su operativni sistemi kompanije Majkrosoft, namenjeni poslovnom okruženju, sadržali video igru pasijans. Uloga ove igre je da motiviše korisnike da ovladaju korišćenjem miša kao ulaznog uređaja, odnosno da funkcije potrebne za to potpuno izmeste u motorički centar kako bi ih koristili paralelno sa intelektualnim aktivnostima.

Redosled obrade ulaznih impulsa prati navedenu hijerarhiju. Dakle, svako događanje zabeleženo čulnim organima prvo će obraditi motoričko-instinktivni centar, zatim emotivni, a kao poslednji intelektualni. Ova činjenica je posebno važna, s obzirom na to da prva dva centra često daju gotovo potpuno obrađene podatke, odnosno formiranu informaciju intelektualnom centru, a da intelektualni centar zatim samo obrazlaže već formiranu odluku. Na primer, ljudi često donose odluke, ne na osnovu činjenica kojim raspolažu, već na osnovu utiska koji je kod njih izazvala osoba koja dostavlja činjenice. Poznati su i suprotni slučajevi - na primer, kao posledica operacije tumora na mozgu izvesnom pacijentu je prekinuta komunikacija između intelektualnog i prethodnih centara. Posledica je, umesto očekivane potpune racionalnosti, bila potpuna neodlučnost - pacijent je mogao satima da analizira meni u restoranu, ali ne i da donese odluku šta će izabrati.

Jedan od važnih zadataka danas jeste i razlučivanje ljudi od računara (mašina, robota). Sa obzirom na to da roboti imaju daleko razvijeniju mogućnost uporednog mišljenja (intelekt) od ljudi, postoji opasnost da će ona biti zloupotrebljena - na primer, slanje ogromnog broja besmislenih komentara ili registracija ogromnog broja korisničkih naloga na nekom Veb sajtu. Iz tog razloga potrebno je razviti efikasne sisteme za utvrđivanje da li je druga strana u interakciji sa našim programom čovek ili mašina. Nasuprot razvijenosti intelektualnog aparata, motorički i emotivni kapaciteti računara danas neuporedivo zaostaju za čovekovim (uporedite najnaprednijeg robota firme *Boston Dynamics* sa umetničkim klizanjem). Ta karakteristika se često koristi u *CAPTCHA* sistemima - u vidu zahteva za prepoznavanjem simbola ili objekata na slici, pitanjima vezanim za podrazumevanu emotivnu reakciju i sličnim.

Prilikom osmišljavanja efikasnog korisničkog interfejsa cilj je što više iskoristiti mogućnosti i karakteristike motoričkog i emotivnog centra, kao moćnijih od intelektualnog. Korišćenje miša predstavlja dobar primer ove prakse, gde su nizovi instrukcija za koje bi bio zadužen intelektualni centar zamenjeni jednostavnim, "nesvesnim" pomeranjem šake i prstiju. Dodatno, često se reakcije elemenata korisničkog interfejsa na akcije korisnika kreiraju tako da stimulišu percepciju nižih centara - na primer animacije. Kompanija Majkrosoft je za kreiranje pratećih zvukova kod interakcije sa korisnikom, za Vista operativni sistem, angažovala gitaristu Roberta Fripa, kao odličnog poznavaoa ovih principa.

Osnovu pojma “opažanje” čini imenica “pažnja”. Danas postoji veliki broj definicija pažnje: čin ili stanje usmeravanja uma na određeni objekat, stanje povećane budnosti, usredsređenost na ograničen broj objekata u čulnom polju ili imaginaciji, itd. Za pravilno razumevanje suštine pažnje prvenstveno je potrebno razumevanje prethodno definisane “tricentričnosti” čoveka, odnosno nivoa na kojima opažaji mogu raditi.

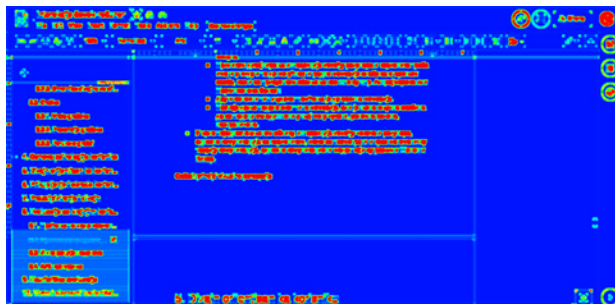
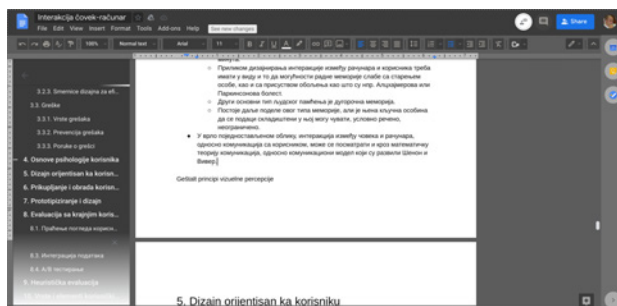
Dalje, jedna od pomenutih definicija uključuje i pojam “budnosti”, pa se za čoveka često vezuju dva osnovna stanja - stanje sna i stanje budnosti (pažnje). Ovakva podela je veoma neprecizna, odnosno gruba, pošto je spektar mogućih stanja mnogo širi. Na primer, postoji stanje dubokog sna, stanje sanjanja, pospanosti, budnosti, uzbuđenosti, itd. Jedna od karakteristika stanja najintenzivnije, duboke pažnje jeste i fotografsko pamćenje, koje je verovatno svaki pojedinac iskusio barem nekoliko puta u životu. U nekim psihološkim modelima se intenzitet pažnje vezuje za raspoloživost intelektualne energije, gde nedostatak iste vodi ka pospanosti (zevanje) a višak ka smejanju.

Sa obzirom na to da duboka pažnja nije ni često, a ni prirodno stanje čoveka, prilikom dizajniranja interakcije sa korisnikom tu činjenicu takođe treba uzeti u obzir. Na primer, autori knjige *Hooked* [26] daju instrukcije, odnosno model dizajniranja interakcije sa korisnikom, koji će dovesti do stvaranja navike korišćenja. Stanje navike, kao znatno manje zahtevno, prirodnije i održivije od stanja duboke pažnje, omogućava da korisnici češće koriste određenu aplikaciju, odnosno da ne traže alternativna rešenja.

U svakom slučaju, prvi korak u interakciji sa korisnikom jeste privlačenje i usmeravanje njegove pažnje. Kod ranih računarskih sistema ovo i nije bio toliko značajan problem, s obzirom na ograničenost korisničkih interfejsa i ponude aplikacija. Danas, posebno na Vebu, privlačenje pažnje korisnika jedan je od kritičnih zadataka. U marketingu je odavno razvijen AIDA model - *Attract, Interest, Desire, Action* [27]. Isti princip važi i kod savremenih Veb stranica - smatra se da ukoliko pažnja korisnika nije privučena primarno prikazanim informacijama (vrh stranice), on će najverovatnije napustiti sajt.

Ljudsko pamćenje je aktuelan predmet istraživanja različitih oblasti - psihologije, neurologije, i drugih. Principi njegovog funkcionisanja još uvek nisu u potpunosti razjašnjeni, ali su ostvareni ogromni pomaci u praktičnom razumevanju njegovog rada, mogućnostima manipulacije i uticajima različitih patologija. Na primer, Dr. Elizabeta Loftus, američki psiholog, u poslednjih nekoliko decenija napravila je ogroman pomak na polju razumevanja koliko je ljudsko pamćenje podložno naknadnoj manipulaciji (preporučujemo njeno izlaganje pod nazivom *How reliable is your memory* na TED-u).

U svakom slučaju, dve vrste ljudskog pamćenja kritično su važne i za interakciju između čoveka i računara. Prvu vrstu čini tzv. kratkoročna, radna memorija. Ova memorija ima ograničen kapacitet i trajnost - obično može da podrži i do sedam objekata u trajanju od par sekundi do par desetina sekundi (nekada do jednog minuta). Prilikom dizajniranja interakcije između računara i korisnika treba imati u vidu i to da mogućnosti radne memorije slabe sa starenjem osobe, kao i sa prisustvom oboljenja kao što su Alchajmerova ili Parkinsonova bolest. Drugi osnovni tip ljudskog pamćenja je dugoročna memorija. Postoje dalje podele ovog tipa memorije, ali je njena ključna osobina da se podaci skladišteni u njoj mogu čuvati, uslovno rečeno, neograničeno.



Slika 38. Određivanje maksimalnog informacionog potencijala korisničkog interfejsa korišćenjem Šenonove entropije.

Na veoma niskom nivou, interakcija između čoveka i računara, odnosno komunikacija sa korisnikom, može se posmatrati i kroz matematički modelovanu teoriju komunikacija, odnosno komunikacioni model koji su razvili Šenon i Viver [28]. Ovaj model je dobar za kvantifikovanje informacionog potencijala sadržaja kojim se komunicira sa korisnikom, odnosno njegove maksimalne vrednosti u teorijskom smislu, zatim otkrivanje prisustva šuma, određivanje prostorne distribucije informativnih potencijala, i drugi. Međutim, emitovani signali (vizuelni, zvučni) nakon prijema prolaze kroz složene procese

obrade i izdvajanja podataka i informacija, tako da je analizu komunikacije neophodno raditi i na višim nivoima apstrakcije.

4.4. Geštalt principi vizuelne percepcije

Percepcija, kao mehanizam koji određuje kako se informacija dobija iz spoljašnjeg i pretvara u doživljaje, je izrazito kontekstualna. Percepcija okruženja ne mora nužno da odgovara njegovom pravom opisu. Čovek u velikoj meri opaža ono što unapred očekuje da opazi. Konkretno, percepciju čoveka sa aspekta očekivanja vode tri faktora [29]:

1. Prošlost, odnosno iskustvo iz identičnih ili sličnih situacija,
2. Sadašnjost, određena tekućim kontekstom situacije koji čine fizičko, socijalno i organizaciono okruženje,
3. Budućnost, definisana postojećim ciljevima u datom trenutku.

Principi se podjednako važno odnose i na korišćenje tehnologije [29]. Vizuelna percepcija čoveka podrazumevano formira strukturu od ulaznih informacija i vidi celine (oblike, figure i objekte) umesto delova (ivice, oblasti, linije). Reč za figuru ili oblik na nemačkom jeziku je *gestalt* pa se tako pravila koja opisuju ovaj složeni mehanizam nazivaju Geštalt principi vizuelne percepcije [30]. Principima se vode vizuelni sistemi čoveka kako bi organizovali, odnosno grupisali elemente u smislene celine. Drugim rečima, kada su elementi organizovani u grupama koje definišu objekat čovek opaža grupu a ne individualne elemente. Principi su urođen mehanizam optimizacije čovekovog mozga u cilju smanjenja kognitivnog naprezanja. Kao takvi su našli i primenu u dizajnu GKI.

Principi značajni za dizajn korisničkog interfejsa obuhvataju:

- Princip blizine (eng. *Proximity*),
- Princip sličnosti (eng. *Similarity*),
- Princip kontinuiteta (eng. *Continuity*),
- Princip ograničavanja (eng. *Closure*),
- Princip simetrije (eng. *Symmetry*),
- Princip figura/pozadina (eng. *Figure/Ground*),
- Princip pokreta (eng. *Common Fate*).

Sa stanovišta dizajna korisničkog interfejsa primena principa može obezbediti logičku strukturu informacija sa aspekta korisnika. U isto vreme pomažu dizajnerima i inženjerima da pravilno grupišu sadržaje i poboljšaju upotrebljivost interfejsa. U nastavku će biti opisan svaki od principa sa naglaskom na dizajn korisničkog interfejsa.

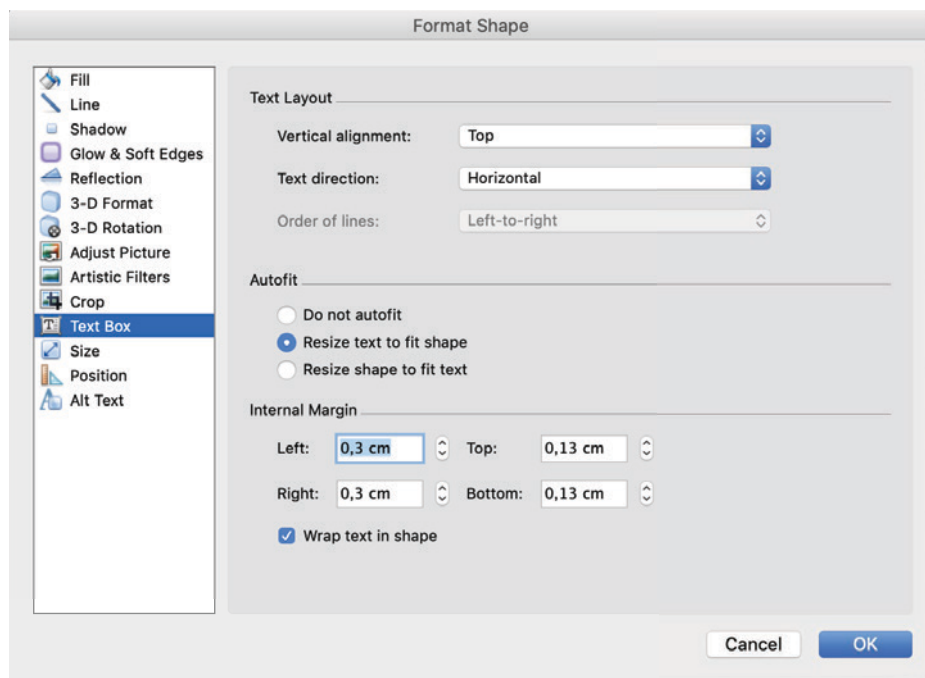
4.4.1. Princip blizine

Međusobna distanca između objekata na interfejsu utiče na to da li su i kako objekti opažaju kao podgrupe. Objekti koji su bliži jedni drugima se se opažaju kao grupa ili celina, dok se međusobno udaljeniji objekti opažaju kao odvojeni.



Slika 39. Princip blizine. Na levoj strani se opažaju redovi, dok se na desnoj strani opažaju kolone. Preuzeto i adaptirano iz [29].

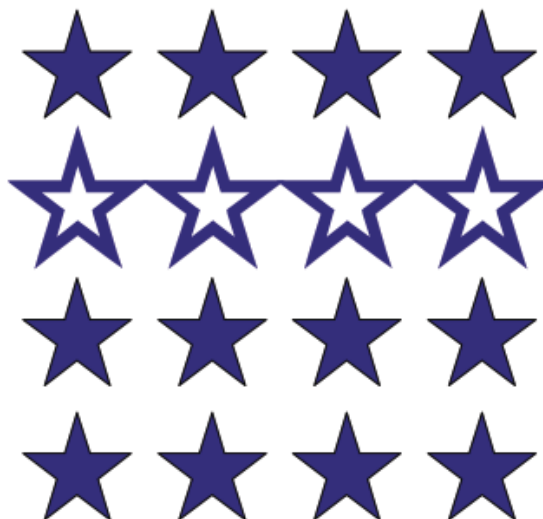
Princip se koristi u definisanju u rasporedu kontrola korisničkog interfejsa (eng. *layout*), kao što su forme sa podacima ili kontrolni paneli sa komandama u različitim aplikacijama. Grupe kontrola se često formiraju tako što se smeštaju u okvire (eng. *bounding box*) ili se grupe razdvajaju linijama za odvajanje (eng. *line separator*).



Slika 40. Princip blizine - Dijalog za formatiranje oblika alata *Microsoft PowerPoint* koristi okvire i horizontalne linije za razdvajanje kako bi komande bile opažane kao logičke celine koje obavljaju specifične funkcije. Logički povezane komande su bliže jedna drugoj tako da se opažaju kao celine. Na primer, raspored teksta, prilagođavanje teksta, podešavanja margina i dugmad za promene, na desnoj strani dijaloga.

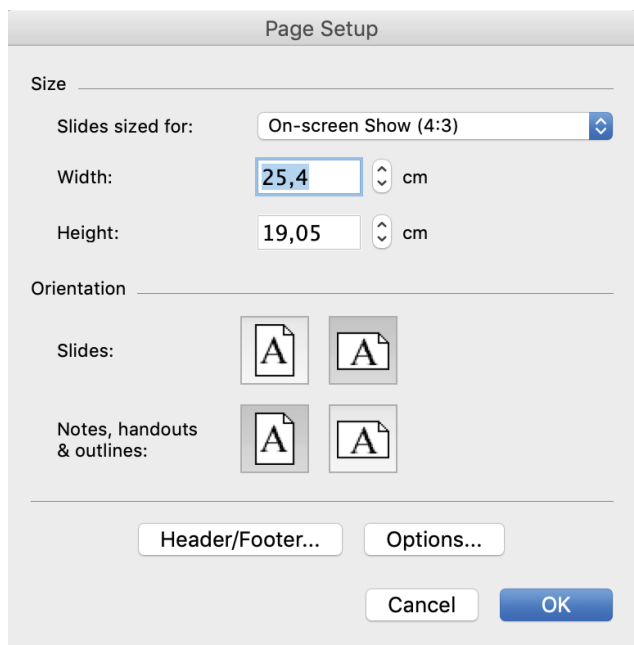
4.4.2. Princip sličnosti

Objekti koji dele vizuelne karakteristike kao što su oblik, veličina, boja, tekstura ili orijentacija se opažaju kao celina. Na taj način se simboli zvezde bez obojene unutrašnjosti opažaju kao grupa (Slika 41).



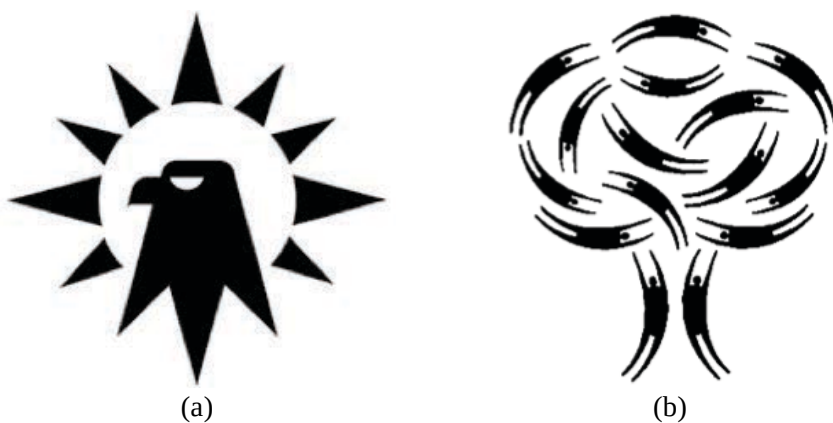
Slika 41. Princip sličnosti - zajednička osobina na osnovu kojih se objekti opažaju kao grupa je boja unutrašnjosti oblika (eng. *fill*). Preuzeto i adaptirano iz [29].

Dijalog za podešavanje stranice (Slika 42) kombinuje principe blizine i sličnosti za formiranje grupa - ikone za orijentacije koriste identične simbole i opažaju se kao zasebna grupa; padajuće liste za podešavanja veličine koriste simbole strelica i imaju slične oblike; dugmad za dodatne opcije i zatvaranje dijaloga se opažaju kao odvojene grupe iako imaju slične oblike zbog toga što su udaljenije. Horizontalne linije za razdvajanje dodatno odvajaju grupe opcija.



Slika 42. Princip sličnosti u dizajnu dijaloga za podešavanje izgleda stranice.

Princip sličnosti se koristi i u dizajnu simbola ikona (Slika 43). Ikone na slici se sastoje od delova identičnih ili sličnih oblika i boja. Neposredno se opažaju kao celine, dok se delovi uočavaju nakon toga, ukoliko se pažnja dodatno usmeri na strukturu simbola.

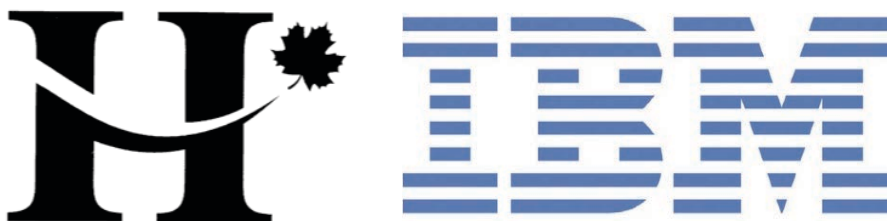


Slika 43. Princip sličnosti u dizajnu simbola sastavljenih od jednostavnijih oblika.

4.4.3. Princip kontinuiteta

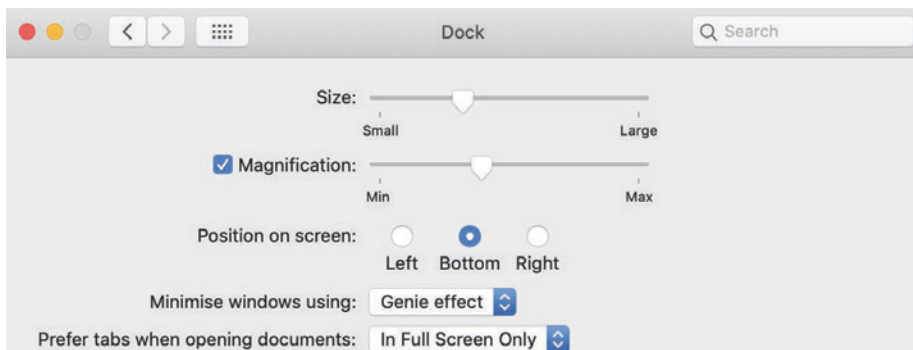
Prethodna dva principa opisuju sklonost percepcije čoveka da opaža i organizuje objekte u grupe. Drugi principi se odnose na našu sklonost da rešavamo dvosmislene opažaje ili dodajemo nedostajuće informacije kako bi objekte doživljavali kao celine.

Kontinuitet kao princip ukazuje da naša vizuelna percepcija vidi kontinualne forme pre nego nepovezane delove. Drugim rečima, čovek ima tendenciju da opaža kontinuitet tako što automatski dodaje ili predviđa nedostajuće delove kako bi se sklopila celina (Slika 44).

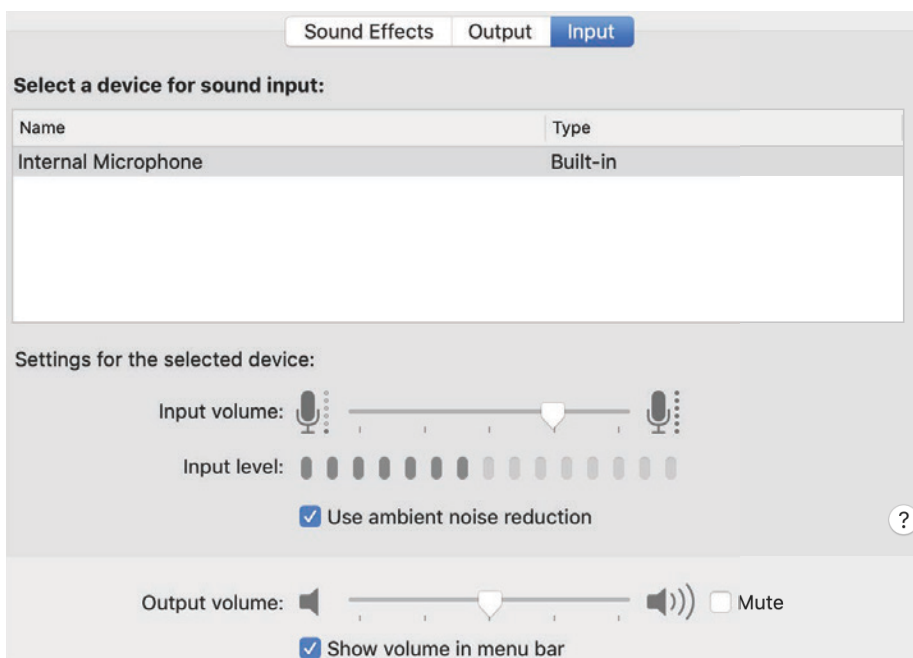


Slika 44. Princip kontinuiteta - (levo) simbol koji se sastoji od delova koji nisu spojeni se opaža kao slovo H; (desno) logo poznatog brenda se opaža kao akronim kompanije iako se sastoji od nepovezanih pravolinijskih segmenata. Izvor: By Paul Rand - Captured from the front page of the IBM Notice of 2007 Annual Meeting and Proxy Statement.

Primer u korisničkim interfejsima jeste klizač (eng. *slider*) kao što prikazuje Slika 45. Klizač se opaža kao jedan opseg ili skala čiju trenutnu vrednost definiše pozicija pokretne ručice (eng. *handle*), a ne kao dva opsega koji su razdvojeni ručicom.



(a)



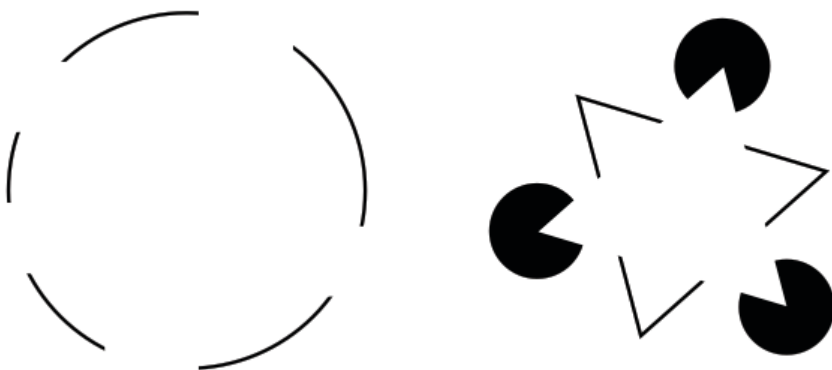
(b)

Slika 45. Princip kontinuiteta - Klizač kao kontrola korisničkih interfejsa koja se koristi za definisanje vrednosti specifičnih parametara u podešavanjima.

4.4.4. Princip ograničavanja

Ograničavanje je princip vizuelne percepcije koji opisuje čovekovu sklonost da automatski dopunjava ili zatvara granice otvorenih formi da bi se opažala celina a ne segmenti. Na levoj strani (Slika 46) se opaža kružnica umesto pojedinačnih lukova. Na desnoj strani se prazan prostor opaža kao objekat - ono što vidimo

jeste beli trougao koji preklapa trougao ispod i tri crna kruga, iako su delovi zapravo tri V oblika i tri crna kruga sa odsečcima.



Slika 46. Princip ograničavanja - nepovezani segmenti se opažaju kao celina.

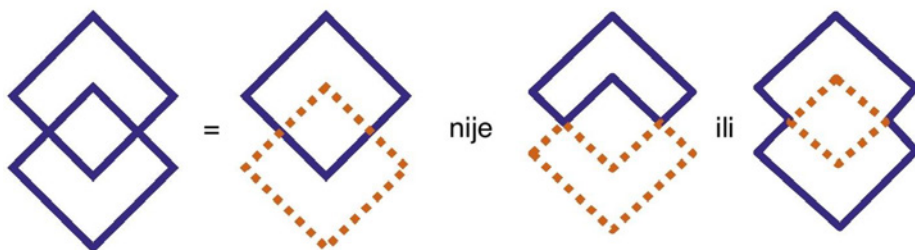
Ograničavanje se koristi u dizajnu simbola ikona za predstavljanje akcija u Veb korisničkim interfejsima [11]. Segmenti i delimično vidljivi objekti se opažaju kao jedan, složeniji simbol.



Slika #. Princip ograničavanja u dizajnu simbola ikona. Izvor [11].

4.4.5. Princip simetrije

Vizuelna percepcija čoveka obrađuje složenije scene tako što grupiše jednostavnije, simetrične oblike u cilju smanjenja kompleksnosti. Drugim rečima, podaci dostupni vidnom polju se u opštem slučaju mogu interpretirati na veći broj načina, međutim vizija čoveka automatski strukturiira opažene podatke kao simetrične reprezentacije kako bi ih pojednostavila. Slika 47 prikazuje objekat koji se opaža kao dva preklapajuća kvadrata, a ne kao dva poligona u nastavku. Razlog tome je što je prvi oblik jednostavniji od dva naredna - ima manji broj strana i izraženiju simetriju.



Slika 47. Princip simetrije - objekat na slici opažamo kao prvu reprezentaciju na desnoj strani pre nego dve koje slede zbog toga što je jednostavnija. Preuzeto i adaptirano iz [29].

U vizuelnom dizajnu i računarskoj grafici princip se koristi za reprezentaciju trodimenzionalnih scena u dvodimenzionalnom prostoru (Slika 48).



Slika 48. Princip simetrije kod opažanja trodimenzionalnih scena u dvodimenzionalnom prostoru. Izvor: Naslovna strana knjige Coherence in Thought and Action (Thagard, 2002).

4.4.6. Princip figura/pozadina

Ovaj princip opisuje način na koji percepcija čoveka opaža scene kod kojih su objekti raspoređeni po dubini prikaza, odnosno po zamišljenoj osi normalnoj na ravan displeja. Čovek razdvaja vizuelno polje kao figuru (sadržaj koji je u prvom planu) i pozadinu (sadržaj u drugom planu). Objekti koji pripadaju figuri su predmet pažnje, dok objekti pozadine nisu.

Dva faktora utiču na način na koji vizuelna percepcija deli scenu na figuru i pozadinu:

- osobine same scene, odnosno načina njenog dizajna i uređenja (faktor dizajna),
- fokusa pažnje posmatrača (faktor korisnika).

Na primer, Slika 49 prikazuje manji objekat koji preklapa veći objekat i zbog toga se objekat u prvom planu (krug) opaža kao figura dok se kvadrat vidi kao pozadina.



Slika 49. Princip figura/pozadina - kada se objekti preklapaju, potpuniji se opaža kao figura.

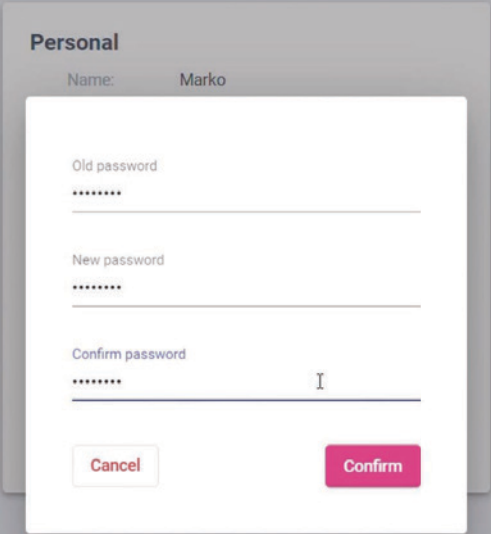
Čovekova percepcija figure i pozadine zavisi i od toga gde se svesno usmerava pažnja pri posmatranju scene. Umetnik Mauris Kornelis Ešer (eng. *Maurits Cornelis Escher*) je koristio ovaj fenomen i kreirao dvosmislene slike u kojima se figura i pozadina naizmenično menjaju uz zavisnosti od toga gde je pažnja usmerena [31].

U dizajnu korisničkih interfejsa princip se može realizovati tako da se pažnja korisnika usmerava na pozadinu, iza primarnog sadržaja. Dakle pozicije figure i pozadine nisu apsolutne, nego mogu biti relativne - sadržaj pozadine može privući pažnju korisnika i biti u prvom planu. U ovom slučaju, pozadina korisniku saopštava temu, motiv ili poruku aplikacije ili Veb sajta (Slika 50).



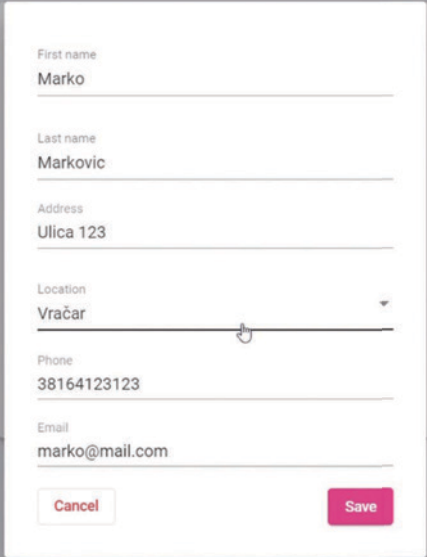
Slika 50. Princip figura/pozadina - sadržaj u pozadini Veb sajta može privući pažnju korisnika i zapravo predstavljati figuru. Izvor: Gorska služba spasavanja Srbije, <http://gss.rs> (15.11.2020).

Princip je implementiran u korišćenju iskaćućih (eng. *popup*) elemenata korisničkog interfejsa kao što su meniji, forme sa podacima i dijalozi sa porukama (Slika 51).



A modal form titled "Personal" is displayed over a grey background. The form contains three password input fields labeled "Old password", "New password", and "Confirm password". Each field has a masked password of seven asterisks. At the bottom of the form are two buttons: "Cancel" and "Confirm".

(a)



A modal form for updating user information is shown over a grey background. It includes input fields for "First name" (Marko), "Last name" (Markovic), "Address" (Ulica 123), "Phone" (38164123123), and "Email" (marko@mail.com). The "Location" field is a dropdown menu currently showing "Vračar". At the bottom are "Cancel" and "Save" buttons.

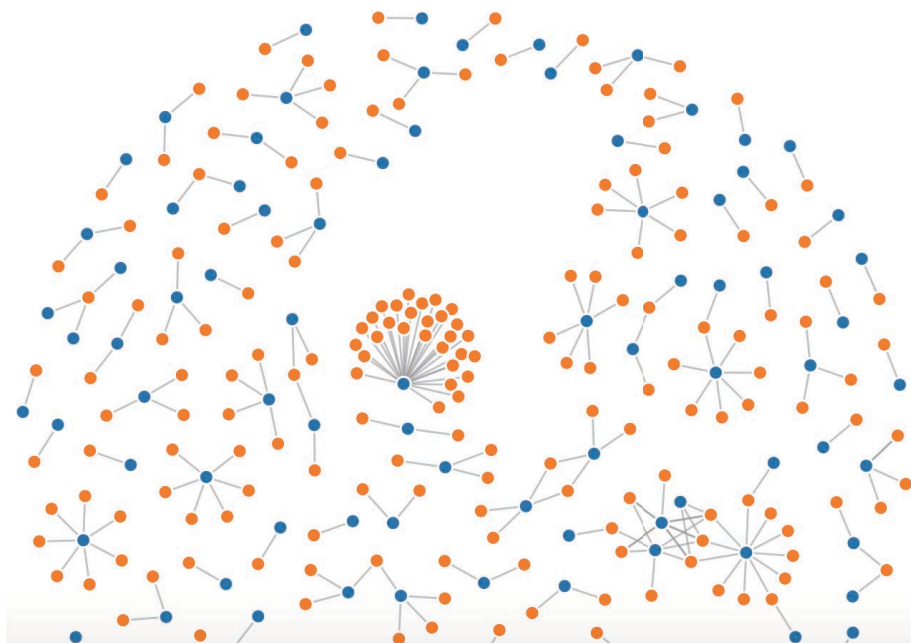
(b)

Slika 51. Princip figura/pozadina - modalne forme za ažuriranje podataka se mogu dodatno istaći tako što čine pozadinu aplikacije nedostupnom.

4.4.7. Princip pokreta

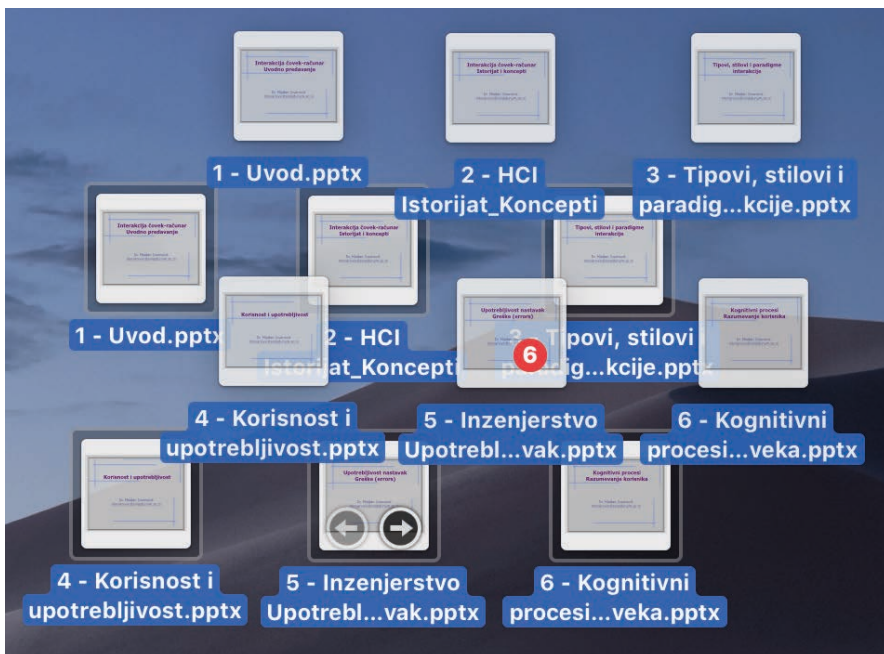
Dok se prethodni principi odnose na statičke scene u kojima se objekti ne pomeraju, princip pokreta se odnosi na dinamičke scene u kojima se objekti na neki način kreću. Princip navodi da se objekti koji se kreću zajedno opažaju kao grupa.

Ovaj princip se koristi u korisničkim interfejsima aplikacija koje rade sa podacima kako bi se vizuelno razdvojile grupe podataka na osnovu različitih parametara. Logički povezane grupe podataka (poznate kao klasteri) pored pokreta u istom pravcu ili smeru, mogu deliti i vizuelne karakteristike (kao što su oblik, boja, veličina i orijentacija) kako bi bile još uočljivije (Slika 52).



Slika 52. Princip pokreta - Aplikacije za vizuelizaciju podataka omogućavaju selekciju i prevlačenje grupe izborom proizvoljnog čvora klastera grafa. Čvorovi klastera se kreću u pravcu prevlačenja i opažaju kao grupa. Izvor: Disjoint Force-Directed Graph, D3js, <https://observablehq.com/@d3/disjoint-force-directed-graph> (15.11.2020).

Prevlačenje objekata na interfejsu takođe predstavlja primer principa pokreta. Zajedničko vizuelno isticanje (eng. *highlighting*) i pokret čine da se objekti opažaju kao grupa (Slika 53). U ovom slučaju se koristi i princip sličnosti zbog toga što objekti izgledaju identično (simboli za prezentacije).



Slika 53. Principi pokreta i sličnosti kod prevlačenja objekata na korisničkom interfejsu.

5. Dizajn orijentisan ka korisniku

Dizajn interaktivnog proizvoda je kreativna i primenjena aktivnost čiji je cilj razvoj proizvoda koji će korisnicima omogućiti ispunjenje njihovih ciljeva. Ciljevi mogu biti vezani za profesionalni ili privatni život korisnika u zavisnosti od namene sistema. U prethodnim poglavljima su opisani principi korišćenja interaktivnih sistema kako bi bolje razumeli potrebe korisnika poput korisnosti, upotrebljivosti i tipova i stilova interakcije, kao i kognitivni procesi koji se tada odigravaju kroz relevantne koncepte psihologije korisnika.

U ovoj sekciji će biti obrađen proces dizajna interaktivnog proizvoda. U razvoju softvera, modeli i kod predstavljaju pre svega sredstvo komunikacije sa drugim programerima ili inženjerima koji poseduju dovoljno stručnih znanja i veština kako bi razumeli predmet komunikacije. Sa druge strane, korisnički interfejs je sredstvo komunikacije sa krajnjim korisnicima, koji pored toga što ne moraju biti inženjeri, imaju specifične navike, potrebe i zahteve. Zbog toga je osnovna pretpostavka dizajna interaktivnog sistema aktivno učešće korisnika u toku samog procesa, od prikupljanja korisničkih zahteva, preko dizajna i implementacije, do testiranja sa korisnicima. Razlog tome jeste spoznavanje i definisanje stvarnih potreba korisnika i njihova pravilna realizacija u budućem sistemu.

Ovakav pristup sa druge strane donosi i niz problema u vezi sa načinom na koji se korisnici uključuju u proces dizajna interaktivnog proizvoda. Korisnici nisu dizajneri, u opštem slučaju nemaju potrebno znanje i veštine, i ne mogu precizno definisati kako korisnički interfejs treba da izgleda i funkcioniše. Pored toga mogu imati specifične i raznovrsne potrebe. Zbog toga ih je neophodno angažovati na odgovarajući način, korišćenjem ustanovljenih i verifikovanih tehnika i metoda koje će biti obrađene. Ukoliko korisnici greše, teže uče, sporo koriste ili imaju drugi problem u korišćenju specifičnog dela interfejsa, to je znak da je problem u dizajnu interfejsa, ne u korisniku.

5.1. Faze i karakteristike dizajna interaktivnog sistema

Kako bi proces davao što je moguće kvalitetnije rezultate i na pravi način angažovao krajnje korisnike, on se sastoji od četiri osnovne faze:

- prikupljanje i definisanje korisničkih zahteva,
- dizajniranje rešenja,
- implementacija funkcionalnih sistema i
- testiranje sa korisnicima.

Nabrojane aktivnost će biti detaljnije obrađene u poglavljima koja slede. U nastavku je dat opštiji opis svake aktivnosti.

Prikupljanje i definisanje korisničkih zahteva predstavlja upoznavanje ciljne grupe korisnika sa aspekta korišćenja sistema koji se kreira. Drugim rečima, potrebno je znati stvarne potrebe naših korisnika koje čine osnovu dizajna i implementacije. Ova aktivnost je ključna u dizajnu i biće razrađena u Poglavlju 6.

Dizajniranje rešenja, odnosno verzije dizajna jeste realizacija, otelotvorenje zahteva iz prethodne aktivnosti kako bi korisnici imali predstavu o tome kako bi budući sistem mogao da izgleda i da se ponaša. Postoje dve vrste dizajna - konceptualni i fizički. Konceptualni dizajn podrazumeva idejno rešenje koje opisuje izgled i funkcije proizvoda na visokom nivou apstrakcije (na primer, papirni crtež budućeg sistema). Fizički dizajn konkretizuje konceptualni dizajn tako što dodaje detalje kao što su elementi korisničkog interfejsa i interaktivnost kako bi korisnici imali bogatiji doživljaj sistema. Oba tipa dizajna se mogu realizovati u većem broju iteracija i alternativa. Ove aktivnosti će biti detaljnije opisane u Poglavlju 7.

Implementacija funkcionalnih sistema podrazumeva kreiranje interaktivnih verzija sistema, odnosno sistema koji predstavlja finalnu verziju proizvoda. Ova aktivnost realizuje sistem u nekom od programskih jezika. Biće opisana arhitektura, ulazno-izlazni mehanizmi i standardne kontrole GKI zajednički za različite tehnologije (Poglavlje 10).

Testiranje sa krajnjim korisnicima je procena upotrebljivosti i prihvatanja (eng. *acceptance*) dizajna ili sistema. Procena može biti kreirana i izražena korišćenjem različitih kriterijuma kao što su broj grešaka koje korisnik pravi, nivo ispunjenosti korisničkih zahteva, u kojoj meri je sistem privlačan korisniku i drugi. Glavni razlog aktivnog angažovanja korisnika u celokupnom procesu dizajna interakcije je veća verovatnoće isporuke prozroda prihvatljivog korisnicima. Sistem kao celina se može testirati sa različitih aspekata. Na primer, funkcionalno testiranje, testiranje bezbednosti, testiranje zadovoljenja standarda i drugi. Važno je napomenuti da evaluacija sa korisnicima ne zamenjuje druge oblike testiranja, već je njima komplementarna i služi kao dopuna.

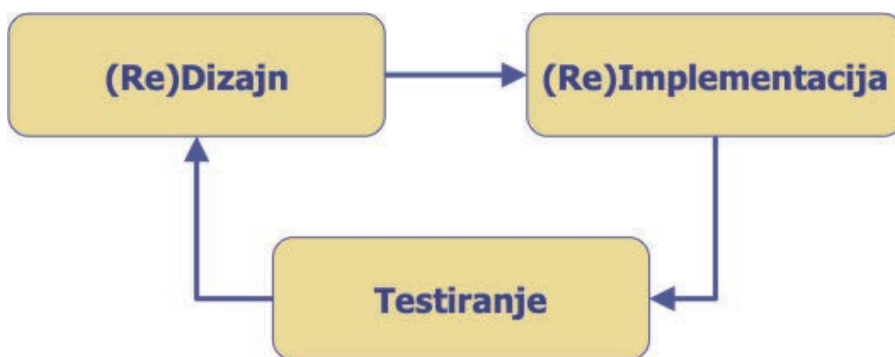
Ključne osobine procesa dizajna interaktivnog proizvoda su:

- akcenat na krajnjim korisnicima,
- konkretni ciljevi upotrebljivosti i doživljaja korisnika, i
- iterativni i inkrementalni karakter.

Akcent na korisnicima podrazumeva aktivno i kontinuirano učešće korisnika u toku svih aktivnosti dizajna interaktivnog proizvoda. Učešće počinje komunikacijom sa korisnicima radi spoznaje njihovih potreba i definisanja zahteva za izradu sistema. Zatim slede rane evaluacije konceptualnih i fizičkih dizajna sistema radi dobijanja pravovremene povratne informacije o sistemu koje utiču na nastavak procesa (u smislu modifikacija dizajna). I na kraju, testiranje funkcionalnog rešenja sa korisnicima pre puštanja sistema u operativnu upotrebu.

Ciljevi dizajna sa aspekta upotrebljivosti i doživljaja korisnika moraju biti definisani pre početka dizajna rešenja i potvrđeni sa krajnjim korisnicima. Jasno definisani ciljevi usmeravaju, kontrolišu i povezuju faze dizajna, implementacije i evaluacije.

Iterativnost i inkrementalnost su osobine čitavog procesa kao skupa faza. Konceptualni i fizički dizajn se u praksi realizuju u ciklusima - rezultati ranih testiranja sa korisnicima se analiziraju i sprovode se modifikacije dizajna na osnovu povratnih informacija od korisnika (Slika 54). Saradnja dizajnera i korisnika počinje diskusijama u vezi sa zahtevima korisnika. Nakon što se zahtevi realizuju prvim prototipima tada se usklađuju pogledi i predstave o tome kako bi budući sistem trebalo da izgleda i funkcioniše. Zbog toga se aktivnosti dizajna ponavljaju kako bi se došlo do što kvalitetnijeg rešenja. Inkrementalni karakter procesa znači da se u svakoj narednoj iteraciji dizajn obogaćuje, kako dodavanjem novih elemenata, tako i modifikacijom postojećih kako bi korisnicima bio što prihvatljiviji. Osobine kao takve su karakteristične i za razvoj softvera [32].

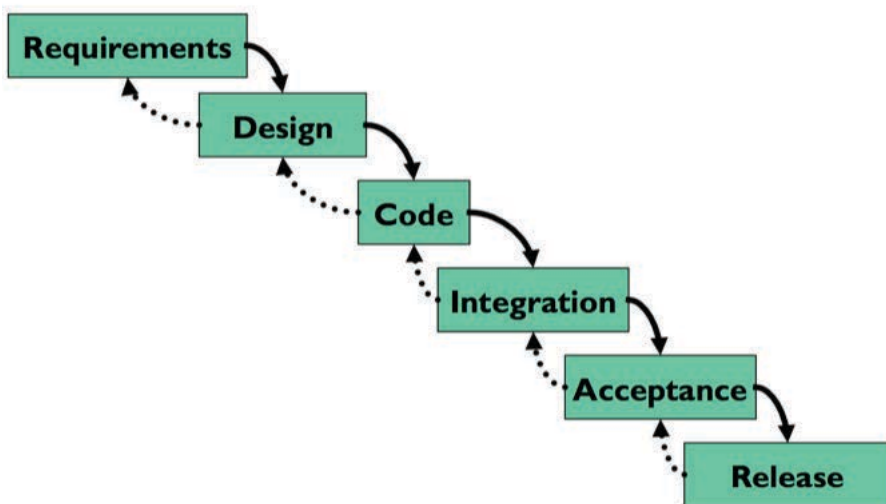


Slika 54. Razvoj interaktivnog proizvoda se odvija u ciklusima (iteracijama).

5.2. Model razvoja softvera

Radi boljeg razumevanja, upoređićemo proces dizajna interaktivnog proizvoda sa postojećim procesima u razvoju softvera. Model Vodopada [33] (eng. *Waterfall Model*), predložen 70-tih godina 20. veka je jedan od prvih modela razvoja softvera i predstavlja osnovu mnogih kasnijih modela razvoja koji se i danas koriste [32]. Model je linearan tako da svaka faza mora biti završena pre početka naredne faze u procesu (Slika 55). Na primer, zahtevi moraju biti definisani pre početka dizajna. Zatim dizajn mora biti kreiran pre implementacije. Implementacija je preduslov za integraciju. Nakon integracije se vrše testiranja od čijih rezultata zavisi da li će softver biti isporučen klijentima.

Osnovna prednost modela je što sistematizuje razvoj i uvodi disciplinu u radu inženjera i timova u smislu poštovanja pravila. Pre svega, u osmišljavanju rešenja (kroz definisanje zahteva i dizajn) pre nego se krene sa implementacijom. Na ovaj način se povećava kvalitet rezultata procesa razvoja uz optimalno korišćenje resursa.



Slika 55. Šematski prikaz Model Vodopada u razvoju softvera. Preuzeto i adaptirano iz Paul Smith's work at Wikipedia, CC BY 3.0.

Faze su povezane na taj način što svaka faza daje specifičan rezultat koji predstavlja potreban ulaz u narednu fazu, kao što su specifikacija zahteva, dizajn modula, implementacije modula, moduli/interfejsi za integraciju, testovi i njihovi rezultati.

Svaka faza ima verifikaciju - dizajn se verifikuje na osnovu zahteva, implementacija se testira na osnovu dizajna i slično. U slučaju da verifikacija ne daje zadovoljavajuće rezultate, prethodna faza se ponovo izvršava kako bi se otklonili nedostaci i sledeći pokušaj verifikacije dao potrebne rezultate.

Model kao takav ima određena ograničenja ako se primenjuje u razvoju interaktivnog proizvoda, odnosno korisničkog interfejsa. Kritični aspekti proizilaze iz faktora krajnjeg korisnika, odnosno zbog toga što su krajnji korisnici retko i nedovoljno uključeni u proces razvoja. Na primer, greška u ranoj fazi, kao što je nedostajući zahtev korisnika, može biti otkrivena relativno kasno u procesu - na primer u fazi testiranja prihvatljivosti. Ovo može zahtevati skupe promene u međufazama u pogledu količine posla i resursa potrebnih za njegovo obavljanje.

Sledeće, zahtevi korisnika se mogu menjati vremenom kao posledica promena poslova koje obavljaju ili okruženja u kojima borave. Rano i kontinuirano prisustvo korisnika u procesu je neophodno da bi ove promene bile realizovane kroz dizajn i implementaciju. Iako u modelu Vodopada postoje ciklusi između faza, učesnici u ovim iteracijama su uglavnom inženjeri različitih profila (dizajneri, programeri, testeri), a ne krajnji korisnici koji se pojavljuju mahom u prvoj i poslednjoj fazi.

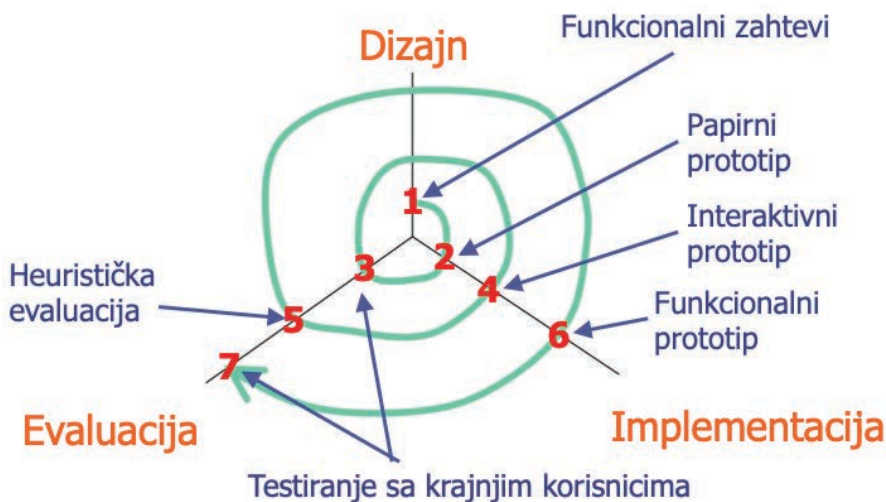
Model Vodopada nije dobro prilagođen i donosi rizike u dizajnu korisničkog interfejsa. Dizajn korisničkog interfejsa složen je zbog toga što je teško predvideti reakcije korisnika u toku dizajna i implementacije konkretnih zahteva ili ideja. Zatim nedovoljno uključivanje krajnjih korisnika u proces razvoja. Kao što je napomenuto, korisnici se pojavljuju u početnoj i završnoj fazi razvoja - prikupljanju i analizi zahteva, i testiranju. Ovaj nivo angažovanja može biti dovoljan za dizajn i realizaciju servisa koje sistem pruža sa aspekta ulaza, logike i rezultata, ali ne i u dizajnu korisničkog interfejsa. Rizik je da interfejs može biti koristan, ali ne i upotrebljiv korisnicima. Na kraju, problemi u korišćenju implementiranog interfejsa mogu usloviti ponavljanje čitavog procesa što je skupo u pogledu dodatnog vremena i angažovanja inženjera. Problemi interfejsa često zahtevaju promene u zahtevima i dizajnu. Promene mogu biti značajne, kao što su odbacivanje pisanog i testiranog programskog koda.

5.3. Spiralni model razvoja korisničkog interfejsa

Korisnički interfejs se realizuje u ciklusima koji se sastoje od dizajna (idejnog rešenja), implementacije (funkcionalnog rešenja), i testiranja sa korisnicima. Na ovaj način se prethodno navedeni rizici u dizajnu interfejsa mogu kontrolisati i redukovati. Na osnovu rezultata evaluacije interfejs se redizajnira,

kreiraju ili modifikuju funkcionalni prototipi, ponovo evaluiraju i proces se ponavlja dok se ne kreira interfejs zadovoljavajuće upotrebljivosti za korisnika. Sa obzirom na planirane i dostupne resurse i vreme, pre početka čitavog procesa se moraju definisati konkretni ciljevi upotrebljivosti koje finalni proizvod mora da ispuni. Imajući ovo ograničenje u vidu broj iteracija se mora držati pod kontrolom. Drugim rečima, mora se uspostaviti dobar balans između broja iteracija kao uloženi resursa i kvaliteta krajnjeg proizvoda. Ovo se postiže na taj način što se može redukovati broj iteracija i cena pojedinačne iteracije. Primena principa dizajna može smanjiti broj iteracija kreiranjem kvalitetnijih interfejsa. Zbog toga će principi biti obrađeni u Poglavljima 9 i 10. Prototipiziranje interfejsa i tehnike evaluacije koje ne zahtevaju krajnje korisnike (heuristička evaluacija) smanjuju cenu iteracije. Prototipiziranje će biti obrađeno u Poglavlju 7, dok će heuristička evaluacija biti opisana u Poglavlju 9.

U praksi, iterativni i inkrementalni model razvoja korisničkog interfejsa zapravo prerasta u formu spirale (Slika 56). Proces se sastoji od ciklusa, gde su početni ciklusi jeftiniji - radijus spirale definiše cenu iteracije odnosno verodostojnost (eng. *fidelity*) tekuće verzije interfejsa. Proces počinje tako što se na osnovu funkcionalnih zahteva za sistemom (1) kreira papirni prototip budućeg interfejsa sistema (2). Verodostojnost ovakvog prototipa je niska, on predstavlja grubu skicu finalnog interfejsa. Sa druge strane, cena njegove realizacije je niska i kao takav može poslužiti za početno testiranje za korisnicima. Na primer, da li su koncept i struktura interfejsa odgovarajući. Dobijanje rane povratne informacije od korisnika i otkrivanje problema u početku razvoja (3) su jako važni za dalji tok procesa. Informacija i početni prototip se dalje mogu koristiti za realizaciju interaktivnog, digitalnog prototipa (4) korišćenjem alata za prototipiziranje kao što su MockPlus [34] ili Marvel [35]. Alati se lako uče i brzo koriste. Njihovo korišćenje ne zahteva poznavanje programskih jezika i samim tim su prototipi interfejsa u ovoj fazi jeftini za izradu. Kreirani prototip se može verifikovati analitički, bez krajnjih korisnika, već u odnosu na generalizovane kriterijume za ocenu upotrebljivosti korisničkog interfejsa poznate kao heuristike upotrebljivosti [36]. Ovaj tip evaluacije upotrebljivosti je poznat kao heuristička evaluacija (5). Na osnovu procene prototipa u odnosu na heuristike, neki delovi dizajna se zadržavaju kao takvi ili modifikuju, dok se neki uklanjaju. Nakon ovih odluka može se pristupiti izradi funkcionalnog rešenja (6) u nekoj od tehnologija za vizuelni dizajn i programiranje korisničkih interfejsa [37]. Implementirani interfejs se zatim testira sa krajnjim korisnicima na osnovu definisanih ciljeva upotrebljivosti (7).



Slika 56. Spiralni model razvoja korisničkog interfejsa - struktura i aktivnosti.

Ključna prednost spiralnog modela kao pristupu realizaciji korisničkog interfejsa je što eliminiše rizike prevelikog ulaganja u dizajn koji možda neće odgovarati potrebama korisnika.

Početne iteracije koriste jeftine prototipe sa malo ulaganja, dok kasnije iteracije kreiraju sadržajnije i skuplje prototipe.

Rizik je najveći na početku procesa zbog toga što se o korisnicima zna najmanje, odnosno o njihovim očekivanjima, potrebama i stavovima vezanim za korišćenje interfejsa. Zbog toga se u početku izrađuju papirni prototipi koji ne iziskuju veće napore i mogu se lakše modifikovati ili čak odbaciti. Nakon nekoliko dizajna i evaluacija upoznajemo bolje naše korisnike, i modifikacije interfejsa se progresivno smanjuju. Spiralni model predstavlja efikasan i efikasan način implementacije funkcionalnog rešenja koje je korisno i upotrebljivo za korisnike.

U praksi, u jednom trenutku je potrebno završiti sa ciklusima i korisnicima isporučiti interaktivni proizvod. Ne postoji univerzalno pravilo, odnosno merilo kada je to potrebno učiniti. Preporuka je da treba razmotriti tri glavna faktora pre donošenja odluke:

- Raspoloživost resursa za dalje iteracije, pre svega vremena i inženjerskog osoblja.
- Step en ispunjenosti kriterijuma upotrebljivosti definisanih na početku procesa razvoja.

- Stavovi i zadovoljstvo korisnika na osnovu povratnih informacija iz obavljenih testiranja.

U nastavku su dati primeri ranog, papirnog prototipa iz koraka 2 Slika 56 (Slika 57a) i interaktivnog, digitalnog prototipa iz koraka 4 Slika 56 (Slika 57b). Prototipiziranje će biti obrađeno u Poglavlju 7.



(a)



(b)

Slika 57. Spiralni model razvoja korisničkog interfejsa - (a) papirni (materijalni) prototip i, (b) digitalni (računarski) prototip.

Važno je istaći da spiralni model nije zamena za druge modele razvoja softvera, već je komplementaran i može se koristiti u kombinaciji sa postojećim modelima.

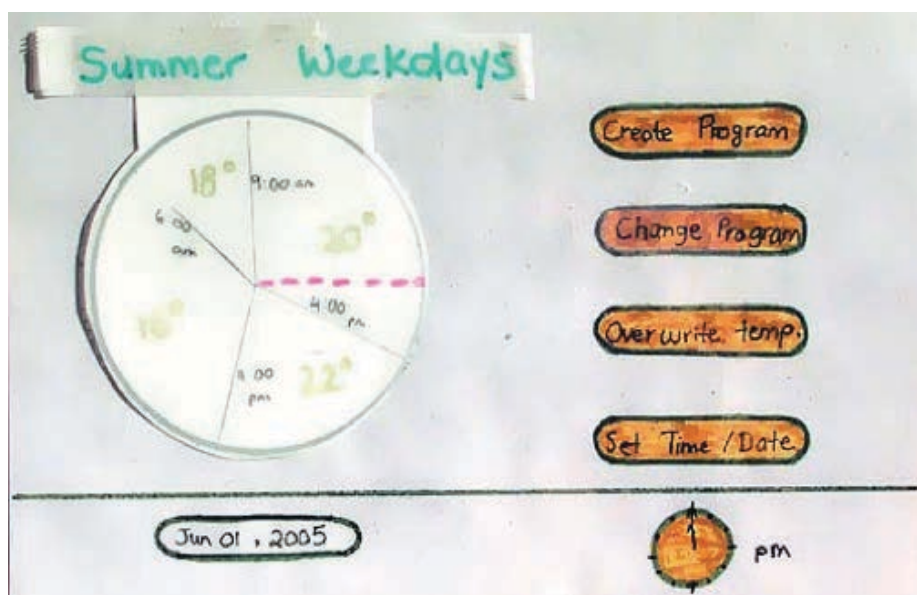
Jedna od tehnika koja se primenjuje u izradi ranih prototipa jeste princip alternativa dizajna. U nekim situacijama mogu postojati višestruka rešenja za realizaciju jedne ideje u smislu korisničkih interfejsa različitih dizajna. Tada se umesto rane odluke za jedno rešenje, preporučuje dizajn većeg broja rešenja koja se potom mogu testirati sa korisnicima kako bi se izabrao najpogodniji dizajn koji bi se dalje razrađivao.

Sa aspekta krajnjeg korisnika, alternative dizajna donose dve značajne prednosti.

Ljudi su generalno bolji u poređenju većeg broja stvari nego u vrednovanju jedne stvari u izolaciji. Shodno tome, korisnici generalno daju konstruktivnije kritike ako poredе alternative dizajna jedne ideje [38].

Slično tome, predlaganje jednog rešenja može formirati neobjektivan stav kod predlagača-dizajnera prema sopstvenom radu kao posledica uloženog truda. U isto vreme, korisnici mogu oklevati da kritikuju jedno ponuđeno rešenje u slučaju da ustanove probleme u dizajnu [38].

Slika 58 prikazuje alternative papirnog prototipa korisničkog interfejsa za kontrolu kućnog klima uređaja zasnovane na različitim metaforama.



(a)

Program

	from	to	temperature
Morning	7:00	9:00	15
Day	7:00	5:00	15
Evening	5:00	12:00	15
Night	12:00	7:00	15

Date

Time

Temperature

(b)

Today:

Time: 12:00

Time

Temp.

Program

Time / Date

Change Temp.

(c)

Slika 58. Alternative dizajna korisničkog interfejsa za kontrolu klima uređaja - (a) cirkularni interfejs, (b) tabelarni interfejs, i (c) linearni interfejs. Preuzeto iz [39].

5.4. PACT analiza

Poznato je da dobar dizajn u obzir uzima sledeće faktore:

- Korisnike (eng. *People*),
- Aktivnosti koje će oni obavljati korišćenjem budućeg sistema (eng. *Activities*),
- Kontekst u kojem se odvija interakcija korisnika i sistema (eng. *Context*),
- Tehnologije koje će biti korišćene za interakciju (eng. *Technologies*).

Skup navedenih faktora, odnosno smernica za rezonovanje o dizajnu je poznat pod nazivom PACT analiza [40]. Definiše okvir za definisanje prostora dizajna sa aspekta relevantnih faktora koji zajedno utiču na korišćenje budućeg sistema. Okvir polazi od ideje za dizajn tako što razmatra i konkretizuje faktore kao odgovore na pitanja:

- Ko ? (*People* komponenta),
- Šta ? (*Activities* komponenta),
- Gde ? (*Context* komponenta),
- Kako ? (*Technologies* komponenta).

Okvir podstiče kreativnost u nalaženju rešenja tako što pojednostavljuje proces kroz razmatranje pojedinačnih komponenti. Konkretno, četiri komponente se detaljno analiziraju kroz odgovore na pitanja:

- Korisnici
 - Relevantne osobine i veštine korisnika ?
- Aktivnosti
 - Koje aktivnosti obavlja korisnik ?
 - Koji su ciljevi aktivnosti ?
 - Kako se aktivnosti mogu unaprediti budućim sistemom ?
- Kontekst
 - Okruženje u kojem se odvijaju aktivnosti ?
- Tehnologije
 - Koje tehnologije se koriste i koje se mogu koristiti ?

Korisnik (čovjek) je po prirodi složeno biće. Zbog toga korisnika treba analizirati sa aspekta osobina relevantnih za korišćenje sistema koji se kreira. Ako govorimo o osobinama korisnika one se mogu posmatrati po grupama kao što su:

- Motoričke osobine,

- Psihološke osobine,
- Sociološke i kulturološke osobine.

Motoričke osobine obuhvataju uzrast (doba) i mehaničke karakteristike korisnika kao što pokreti ruke, preciznost pokreta i dodira prstiju ruke, držanje tela i način kretanja. Veličina šake i dužina prstiju definišu ograničenje u interakciji za korisničkim interfejsom, odnosno u korišćenje ulazno-izlaznih uređaja. Na primer, ako posmatramo decu ranog uzrasta, osobe srednjeg doba ili starije osobe ove razlike postaju relevantne za dizajn interfejsa. Ovo ima direktan uticaj na izbor vrste elemenata korisničkog interfejsa (dugmad, ikone ili meniji), veličinu izabranih elemenata, i njihovo raspoređivanje na interfejsu. U slučaju da se dizajnira dodirni korisnički interfejs, u razmatranje se moraju uzeti i preciznost pokreta i pritiska prstiju korisnika.

Psihološke osobine su opisane u Poglavlju 4. Pažnja, percepcija i memorija su ključni za efikasno i efektivno korišćenje korisničkog interfejsa. Sa aspekta ovih osobina, korisnički interfejs se može posmatrati ne samo kao sredstvo za ispunjenje specifičnih ciljeva korisnika, već svojim dizajnom upravlja ovim osobinama. Ono što je ovde ključno je da te osobine predstavljaju ograničene resurse čoveka i zbog toga se moraju pažljivo koristiti. Ovaj skup psiholoških osobina je važan za upotrebljivost korisničkog interfejsa. U zavisnosti od toga da li korisnički interfejs svojim izgledom i ponašanjem optimalno koristi resurse modela procesiranja čoveka (Slika 36), korisnik će ulagati manji ili veći napor pri učenju interfejsa, brže ili sporije koristiti opcije interfejsa, i praviti greške čiji broj i ozbiljnost utiče na tok korišćenja interfejsa. Prethodno iskustvo, odnosno naučene procedure u korišćenju sličnih sistema se takođe moraju uzeti u obzir prilikom dizajna korisničkog interfejsa.

Pored ovih osobina, postoje i druge koje više utiču na doživljaj korisnika kao dublju psihološku reakciju na korišćenje sistema. U ovaj skup osobina korisnika spadaju motivacija, stavovi i emocije. Ove osobine su važne za korisničke interfejse određenih vrsta aplikacija. Na primere, tipovi motiva korisnika se moraju razmatrati u aplikacijama za edukaciju i učenje. Sa druge strane, u dizajnu video igara su značajni estetski efekti interfejsa igre i emotivna stanja koje scenariji i mehanizmi igre izazivaju kod korisnika.

Sociološke i kulturološke osobine mogu biti relevantne za dizajn korisničkog interfejsa. U ove osobine spadaju jezik, običaji, navike i obrasci ponašanja, i društvene, kulturne i moralne norme korisnika. One mogu uticati na korišćenje specifične terminologije interfejsa razumljive za korisnika (reči i fraze elemenata korisničkog interfejsa kao što su dugmad, meniji, veze i dijalози sa porukama), izbor simbola ikona koji su poznati ili prihvatljivi korisnicima, i boja, oblika, veličina i rasporeda elemenata.

Aktivnosti korisnika se mogu analizirati na osnovu cilja, vremena, broja učesnika i složenosti.

Aktivnosti su pre svega određene cijevima koji se žele ostvariti korišćenjem interfejsa. One mogu voditi zadovoljenju različitih potreba korisnika. U tom smislu, možemo govoriti o dve grupe aktivnosti:

- Pragmatične aktivnosti pretežno usmerene na ostvarenje specifične, praktične dobiti,
- Hedoničke aktivnosti čiji je primarni cilj ispunjenje prijatnih emocija kao što su zadovoljstvo i sreća.

Aktivnosti se analiziraju sa aspekta vremena na tri načina:

- Vreme potrebno za izvršenje aktivnosti,
- Učestalost aktivnosti, odnosno da li se izvršava redovno ili povremeno,
- Kontinuitet, odnosno da li je aktivnost po prirodi kontinualna ili se može vršiti sa prekidima.

Sa aspekta broja učesnika u jednoj aktivnosti mogu postojati jednokorisničke ili višekorisničke aktivnosti. Višekorisničke aktivnosti su karakteristične za aplikacije društvenih mreža ili kolaborativne aplikacije za aktivnosti poput planiranja, koordinacije i timskog rada na različitim projektima.

Složenost govori da li se aktivnost može podeliti na jednostavnije korake ili akcije, ili predstavlja elementarnu akciju koja se ne može raščlaniti. Ovo utiče na način na koji aktivnost može biti realizovana u interfejsu. Na primer, kao selekcija dugmeta ili stavke menija, ili kao sekvenca povezanih prikaza sa opcijama (eng. *wizard*).

Kontekst sa može posmatrati kao fizičko, socijalno i organizaciono okruženje u kojem se odvija interakcija sa budućim sistemom, odnosno aktivnosti korisnika.

Fizičko okruženje je opisano fizičkom karakteristikama prostora kao što su tip prostora, dimenzije prostora, osvetljenost, vlažnost, temperatura i nivo buke. Na primer, ukoliko se aktivnosti odvijaju u pokretu, na različitim lokacijama i na otvorenom prostoru, odgovarajuća aplikacija će biti prilagođena mobilnim uređajima i koristiće uglavnom vizuelnu interakciju. Sa druge strane, ako je u pitanju zatvoreni, privatni prostor u kojem se obavljaju svakodnevne aktivnosti tada se može koristiti i digitalni glasovni asistent.

Socijalno okruženje je određeno (ne)prisustvom drugih ljudi prilikom obavljanja aktivnosti korišćenjem sistema, sredstvima komunikacije sa drugim ljudima i strukturom komunikacije. Prisustvo može biti fizičko (u istom fizičkom prostoru) ili virtuelno (u deljenom digitalnom prostoru kao što je onlajn sastanak ili soba za razgovor). Sredstva komunikacije mogu biti govor, vizija,

tekst, ili kombinacija. Struktura komunikacije može centralizovana (jedan korisnik posreduje u komunikaciji sa drugim korisnicima) ili distribuirana (korisnici međusobno komuniciraju nezavisno jedni od drugih).

Organizaciono okruženje je definisano ustanovljenim normama koje se moraju poštovati prilikom obavljanja aktivnosti korisnika. Ovo se prvenstveno odnosi na radna ili javna okruženja. Definisano je poslovnom kulturom organizacije ili ustanove, ustaljenim procedurama i podrškom za obavljanje aktivnosti, pravnim normama, i različitim oblicima transfera znanja. Na primer, ukoliko se dizajnira interfejs sistema za bankarsko poslovanje, on mora realizovati aktivnosti u skladu sa postojećim pravilima i za bankarsko osoblje i za klijente. U slučaju sistema elektronske trgovine, korisnički interfejs mora realizovati naručivanje na način sličan u stvarnosti.

Tehnologija definiše način obavljanja aktivnosti korisnika. Može se analizirati preko tipa tehnologije, kao mobilni i/ili stoni uređaj. Zatim vrsta ulaza korisnika kao eksplicitni ulaz (dodirni displej, pokazivač, tastatura, glas, pogled) ili implicitni ulaz (senzori uređaja korisnika ili specijalizovani senzori pokreta tela ili delova tela čoveka). Izbor izlaza interfejsa kao tekstualni, vizuelni (grafika i video), glasovni, osetni ili multimodalni koji kombinuje prethodne. Na kraju, izbor specifične softverske platforme za realizaciju korisničkog interfejsa u pogledu programskog jezika i platforme za izvršavanje programa.

PACT analiza - slučaj korišćenja carpooling aplikacija za studente

Termin karpuling (eng. *carpooling*) označava zajedničko prevoženje privatnim prevozom (automobil) tako da više od jedne osobe putuje. Ovaj način prevoza smanjuje troškove putovanja. Pored toga predstavlja održiv vid prevoza tako što smanjuje zagađenje, gužve u saobraćaju i potrebu za parking mestom.

Budući korisnici aplikacije su *studenti*. Ova grupa ima zajednička interesovanja kao što su pohađanje nastave, predispitne i ispitne obaveze, aktivizam, i zajedničke aktivnosti kao što su okupljanja radi druženja, izlasci i putovanja.

Sa aspekta budućeg sistema, studenti mogu imati dve uloge - *vozači* i *putnici*. Ove uloge nisu isključive, odnosno jedan student može biti i putnik i vozač.

Sa druge strane, studenti se mogu razlikovati po tome da li samo studiraju, ili studiraju i imaju zaposlenje. Zatim se može ispitati koje prevozno sredstvo koriste prilikom putovanja. Na primer, korišćenjem upitnika se može utvrditi u kojoj meri se koristi javni prevoz, automobil, motor ili bicikl. Na osnovu ovih informacija se može utvrditi i odnos broja vozača i putnika.

Primarna aktivnost korisnika su su *putovanja* ili *prevoženje*. Svrha aktivnosti jeste posećivanje lokacija od interesa kao što su Univerzitet, mesto stanovanja, mesta okupljanja i druge lokacije. Aktivnost može biti i *socijalizacija*, odnosno susreti i druženje sa drugim studentima. Cilj aktivnosti jeste povezivanje studenata u toku vremena provedenog zajedno. *Deljenje vožnji* sa drugim studentima se isto može posmatrati kao vrsta aktivnosti. Za ovu aktivnost je važno postojanje sistema razmene kao kompenzacije vozače i putnike. *Koordinacija* između vozača i putnika je sledeća aktivnost korisnika koju sistem mora podržati. Za ovu aktivnost je potrebno definisati i usklađivati vremena polaska/dolaska do mesta od interesa, i lokacije ruta putovanja (polazište, stajališta i odredište). Kreiranje i održavanje *reputacije* korisnika sistema je aktivnost koja se odnosi i na vozače i na putnike. Povratna informacija o vožnjama, odnosno vrednovanje vozača od strane putnika i obrnuto je važna za funkcionisanje na bezbedan način i kao motivacija za korišćenje aplikacije.

Fizički kontekst aplikacije sačinjavaju lokacije od interesa studenata kao što su stajališta ruta prevoza, slušaonice i laboratorije u zgradama fakulteta, restorani i barovi za druženja, javni prostori za okupljanja i drugi. Okruženje u kojem se odvijaju aktivnosti je gradska sredina i aktivnosti se obavljaju u zatvorenim i otvorenim prostorima u pokretu. Na primer, u automobilu, na stajalištima i unutrašnjim prostorima kao što se fakultet i bar. *Organizacioni kontekst* relevantan za odvijanje aktivnosti čine saobraćana infrastruktura i uslovi (semafori, periodi očekivanih gužvi u saobraćaju), prisustvo i korišćenje drugih vidova transporta, i neplanirane situacije kao što su obustave javnog prevoza, radovi na putevima ili vremenske nepogode.

Prilikom izbor *tehnologije* potrebno je analizirati postojeće aplikacije kao što je *BlaBlaCar* (<https://www.blablacar.rs/>). Sa jedne strane, aplikacija realizuje aktivnosti vezane za karpoling. Sa druge strane, vozač nudi prevoz za novčanu nadoknadu i pretežno se koristi za prevoz između udaljenijih lokacija, između gradova i država. Namena aplikacije je drugačija i manje odgovara potrebama studenata. Sa obzirom na aktivnosti i kontekst korišćenja interfejs bi trebalo da bude prilagođen mobilnom uređaju. Pored toga, mogu se definisati i neki funkcionalni zahtevi sistema kao što su razmena poruka u realnom vremenu, fleksibilne rezervacije, izmene rasporeda, i korišćenje mapa za navigaciju, orijentaciju i informacije o saobraćaju. Za potrebe ovih servisa se mogu analizirati i aplikacije za navigaciju kao što su *Google Maps* (<http://maps.google.com/>) i *Waze* (<https://www.waze.com/>).

Pažljivom analizom korisnika, aktivnosti, konteksta i tehnologije, početna ideja se razrađuje u skup konkretnijih korisničkih zahteva na osnovu kojih može početi njena realizacija:

- Uloge - putnik i vozač,
- Motivacija - druženje, pomoć drugima, zabava, reputacija,
- Poverenje - bezbedno za korišćenje, profil kao vidljiv identitet, vrednovanje,
- Kompenzacija - nemonetarna kao što je kafa, slatkiš, parking mesto,
- Rezervacija - mora biti efikasna i fleksibilna za korišćenje,
- Vožnja - informacije o rutama, učesnicima, saobraćaju, statusu.

6. Definisanje korisničkih zahteva

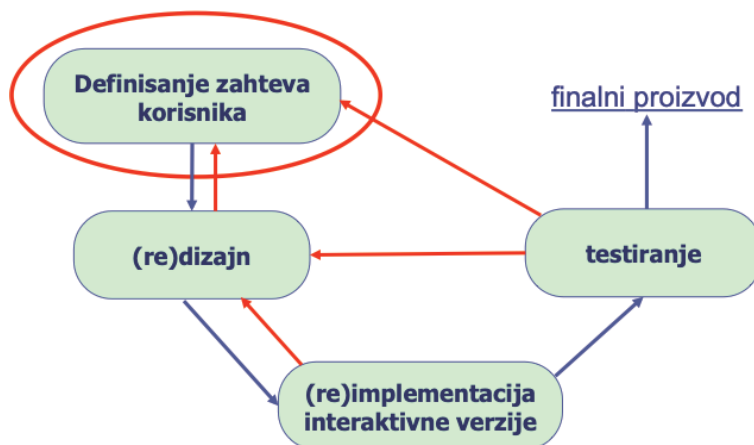
Osnovna pretpostavka uspešnog procesa razvoja interaktivnog sistema jesu pravilno definisani korisnički zahtevi. Zahtevi kao takvi moraju predstavljati stvarne, a ne projektovane ili poželjne potrebe krajnjih korisnika.

Definisanje zahteva je složen proces koji obuhvata prikupljanje, analizu i razumevanje informacija o korisnicima koje su značajne za korišćenje sistema koji se razvija. Ovaj proces ima dva povezana cilja - *razumevanje* potreba korisnika koje sistem treba da zadovolji, i *formulisanje* zahteva za dizajn na osnovu potreba. Ciljevi se realizuju sekvencom aktivnosti, od prikupljanja podataka, preko analize i interpretacije, do artikulisanja zahteva tako da budu razumljivi dizajnerima i inženjerima. Za potrebe ovih aktivnosti razvijene su tehnike za prikupljanje podataka i tehnike za analizu podataka koje će biti obrađene u ovom poglavlju. Važno je napomenuti da se ove aktivnosti mogu odvijati u nekoliko iteracija. Na primer, nakon prikupljanja podataka se može ustanoviti da je potrebno više podataka za analizu, ili da bi se mogli izvesti validni zaključci.

6.1. Vrste korisničkih zahteva

Razumevanje i pravilna primena tehnika prikupljanja podataka i njihove analize je ključna za otkrivanje i korektnu interpretaciju korisničkih zahteva. Inženjerstvo zahteva (eng. *requirements engineering*) je termin koji se koristi da opiše složenost čitavog procesa [41]. U ovom poglavlju će biti obrađeni osnovni elementi procesa neophodni za samostalan rad i dalje usavršavanje.

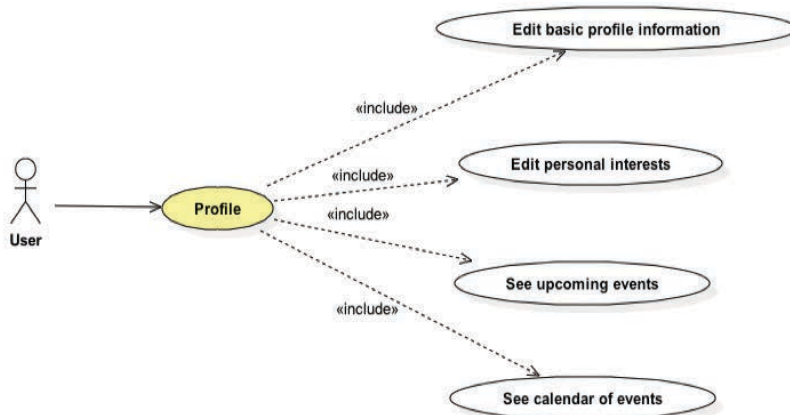
Proces dizajna interaktivnog proizvoda je složen (Slika 59). U fazi dizajna se može ustanoviti da su određeni zahtevi međusobno isključivi ili nedovoljno dobro definisani. Tada se zahtevi ponovo analiziraju. Slično se može desiti u toku faze implementacije ako se utvrdi da se zahtevi kao takvi ne mogu realizovati. U tom slučaju se zahtevi opet analiziraju i redizajniraju odgovarajući elementi rešenja. Na kraju, u toku testiranja se mogu identifikovati nedostajući zahtevi tako da se čitav proces ponavlja.



Slika 59. Definisanje zahteva korisnika u dizajnu interaktivnog sistema.

Postoje različite vrste korisničkih zahteva. Osnovna podela je na funkcionalne i nefunkcionalne zahteve.

Funkcionalni zahtevi govore šta bi sistem trebalo da radi, odnosno koje servise ili funkcije pruža krajnjim korisnicima. Definišu korisnost kao osobinu sistema i njima se primarno bavi razvoj softvera. Istorijski posmatrano, ovo je prva vrsta zahteva koja se pojavila i izučavala. Za potrebe definisanja ove vrste zahteva razvijeni su vizuelni jezici razumljivi i inženjerima i krajnjim korisnicima. UML (eng. *Unified Modeling Language*) je primer takvog jezika (Slika 60).



Slika 60. Funkcionalni zahtevi - UML dijagram slučaja korišćenja (eng. *use case diagram*) se koristi za definisanje funkcija sistema.

Nefunkcionalni zahtevi opisuju druge osobine ili ograničenja sistema. Ovi zahtevi su se javljali uporedo sa razvojem i unapređenjem tehnologije. Krajni korisnici su počeli da se javljaju kao važan faktor korišćenja tehnologija sa aspekta pouzdanosti, bezbednosti, upotrebljivosti i drugih potreba. Različite vrste nefunkcionalnih zahteva se mogu posmatrati i kao različiti načini na koje korisnik vidi funkcije sistema. Na primer, da li je sistem pouzdan, bezbedan, lak za korišćenje, uvažava privatnost korisnika i drugi. Zbog toga se može reći da oni definišu specifične atribute ili kvalitete sistema. Upotrebljivost definiše nefunkcionalne zahteve za korišćenje interaktivnog sistema. Na taj način termini *lako za učenje*, *lako za pamćenje*, *efikasno za korišćenje* i *sigurno za korišćenje* predstavljaju kvalitete korisničkog interfejsa.

Radi boljeg razumevanja razlike između funkcionalnih i nefunkcionalnih zahteva posmatra se razvoj zdravstvenog informacionog sistema.

Za ovaj sistem, primeri funkcionalnih zahteva mogu biti:

- Sistem mora omogućiti lekarima aktivnosti sa pacijentima kao što su pregledi i preporuka lekova.
- Sistem mora omogućiti pacijentima pristup i pregled podataka o pregledima, rezultatima ispitivanja i lekovima.
- Sistem mora omogućiti definisanje rasporeda medicinskog osoblja.
- Sistem mora omogućiti pripremu i izdavanje recepata i drugi.

Nefunkcionalni zahtevi za dati sistem mogu biti:

- Sistem mora biti kompaktan, lagan i prenosiv tako da ga medicinsko osoblje može nositi sa sobom.
- Sistem mora imati efikasnu navigaciju i mogućnost brzog unosa podataka.
- Terminologija aplikacije mora biti jasna i razumljiva.
- Prosečno vreme odgovora sistema mora biti ispod 5 sekundi.
- Sistem mora da spreči neovlašćen pristup podacima korisnika.

6.2. Prikupljanje podataka o korisnicima

Korisnički zahtevi moraju biti potkrepljeni podacima. Prikupljanje podataka je važna aktivnost kako za definisanje zahteva, tako i za evaluaciju sa korisnicima. Tehnike su identične, sa tim što se prikupljaju podaci za različite potrebe - u prvom slučaju radi definisanja zahteva, u drugom slučaju radi procene kvaliteta korišćenja korisničkog interfejsa.

Svrha prikupljanja podataka jeste dobijanje dovoljne količine relevantnih i validnih podataka na osnovu kojih se može formulisati jasan i precizan skup zahteva. Ponekad postoji početni skup zahteva, ali se prikupljaju dodatni podaci kako bi se postojeći skup proširio, pojasnio ili potvrdio. Prikupljanje podataka mora uzeti u obzir različite potrebe koje odgovaraju različitim vrstama zahteva. Na primer, zadaci koje korisnici trenutno obavljaju i njihovi ciljevi, kontekst u kojem se zadaci izvršavaju, i razloge zbog kojih se oni obavljaju na postojeći način. Svi ovi faktori se moraju razmatrati kako bi tehnologija omogućila i unapredila tekući način na koji korisnici obavljaju zadatke.

Broj tehnika za prikupljanje podataka je relativno mali, ali su one same po sebi fleksibilne i mogu se proširivati i kombinovati na različite načine. Pored toga, svaka od njih ima prednosti i kritične aspekte koje treba uzeti u obzir. Ovo donosi različite mogućnosti prikupljanja podataka kako bi rezultujući zahtevi bili što kvalitetniji.

Osnovne tehnike za prikupljanje podataka obuhvataju [12]:

- Analizu dokumentacije ili postojećih rešenja,
- Posmatranje (eng. *observation*),
- Upitnik (eng. *questionnaire*),
- Intervju i fokus grupa (eng. *interview and focus group*),
- Logovanje podataka (eng. *data logging*).

Neke od tehnika, kao što je analiza dokumentacije, ne zahtevaju angažovanje krajnjih korisnika. Dok druge tehnike, poput posmatranja ili intervjuja, zahtevaju aktivno učešće korisnika.

U nastavku će svaka tehnika biti detaljnije opisana.

6.2.1. Analiza dokumentacije ili postojećih rešenja

Formalne procedure i pravila su nekada zapisana u uputstvima. Ona mogu biti izvor podataka o koracima određene aktivnosti ili pravilima koja utiču na izvršavanje aktivnosti. Dokumentacija može biti i neformalne prirode kao što su beleške ili dnevnici aktivnosti korisnika. Ovaj tip dokumentacije može biti značajan za otkrivanje kritičnih aspekata primene formalnih procedura u praksi. Glavna prednost ove tehnike je što ne zahteva angažovanje krajnjih korisnika i daje uvid u normativno definisanje aktivnosti. Sa druge strane, tehniku ne treba koristiti kao jedinu već u kombinaciji sa drugim tehnikama. Dok dokumentacija može biti izvor nekih vrsta nefunkcionalnih zahteva kao što su bezbednost i pouzdanost, ona ne mora biti validan izvor zahteva upotrebljivosti i potreba korisnika uopšte.

Analiza postojećih sistema slične namene je važan izvor podataka. Pre početka realizacije svake ideje neophodno je analizirati postojeća, slična rešenja. Ovo je važno sa dva aspekta.

Pre svega, ideja sama po sebi mora biti inovativna u smislu postojanja razlika i prednosti u odnosu na rešenja zasnovana na sličnoj ideji. U suprotnom slučaju, postoji rizik od ulaganja resursa u razvoj rešenja koja su prethodno realizovana i koje korisnici možda već uveliko koriste.

Pored toga, ukoliko se ustanove novine i prednosti ideje u odnosu na postojeća rešenja, potrebno je pažljivo analizirati rešenja kako bi se identifikovale njihove prednosti i problemi. Prednosti, odnosno pozitivne elemente je potrebno zadržati ili dodatno unaprediti u novom sistemu. Za probleme, odnosno kritične elemente treba ponuditi bolja rešenja u sistemu koji se razvija.

6.2.2. Posmatranje

Korisnici nekada mogu imati poteškoća da neposredno i direktno artikulišu ili opišu ono što rade, odnosno aktivnosti koje obavljaju ili da preciznije objasne kako obavljaju specifične zadatke. U ovom slučaju korisnik se može posmatrati prilikom obavljanja aktivnosti ili zadataka kako bi se došlo do odgovarajućih podataka. Posmatranje podrazumeva boravak sa krajnjim korisnicima, u njihovom radnom ili privatnom okruženju dok obavljaju aktivnosti od interesa. Posmatrač je član razvojnog tima koji provodi vreme se korisnicima, vodi beleške, postavlja pitanja i nadgleda obavljanje aktivnosti. Ova tehnika predstavlja važan izvor podataka o aktivnostima, konkretno za razumevanje prirode zadataka i konteksta u kojem se oni obavljaju. Nivo angažovanja posmatrača varira od neutralnog posmatranja do intenzivnije interakcije sa korisnikom. Sa druge strane, zahteva više vremena i angažovanja, i može rezultovati u velikoj količini dobijenih podataka koja se analizira.

6.2.3. Upitnik

Upitnik je često korišćena tehnika za prikupljanje podataka zbog toga što je poznata većini ljudi. Može se definisati kao skup pitanja kreiranih u cilju prikupljanja specifičnih informacija. Mogu se jednostavno distribuirati ispitanicima, putem elektronske pošte ili na specifičnoj Veb adresi. Najčešće se koriste na daljinu, bez fizičkog prisustva ispitivača.

Dobro dizajnirani upitnici su pogodni za prikupljanje informacija od veće grupe korisnika koja može biti geografski distribuirana tako da neposredan kontakt nije izvodljiv.

Upitnici se razlikuju na osnovu vrste odgovora koju zahtevaju kao:

- Upitnici zatvorenog tipa koji nude predefinisani skup odgovora,
- Upitnici otvorenog tipa čiji su odgovori proizvoljni,

- Mešoviti upitnici kao kombinacija prethodnih.

Glavna prednost upitnika zatvorenog tipa je da je jednostavan za korisnika da odgovori na pitanja. Korisnik ne mora biti naročito artikulisan i ulaže manji napor pri popunjavanju upitnika. Istovremeno je za ispitivača jednostavnije da obradi prikupljene podatke kao izabrane odgovore.

Sa druge strane, korisnik može biti ograničen u pogledu izbora iz predefinisano skupa odgovora. Poznato je da kod ove vrste upitnika korisnici mogu nasumično birati odgovore na šta treba obratiti pažnju prilikom analize rezultata. Može se desiti da u skupu odgovora nisu predviđene opcije koje bi ispitanici želeli da izaberu, odnosno previd detalja odgovora. Ovo može uticati na verodostojnost rezultata u smislu da mogu biti nepotpuni.

Predefinisani odgovori mogu imati dva opšta oblika: da/ne odgovori i odgovori sa skalom (Slika 61).

Da li je sajt singidunum.ac.rs lak za korišćenje ?

Da Ne

(a)

Koliko je singidunum.ac.rs lak za korišćenje ?

vrlo lak 1 2 3 4 5 6 7 veoma složen

(b)

Slika 61. Primeri odgovora upitnika zatvorenog tipa - (a) da/ne odgovor, i (b) odgovor sa skalom.

Ako govorimo o odgovorima sa skalama, postoje dve vrste skala: likert skala (eng. *likert scale*) i semantička diferencijalna skala (eng. *semantic differential scale*).

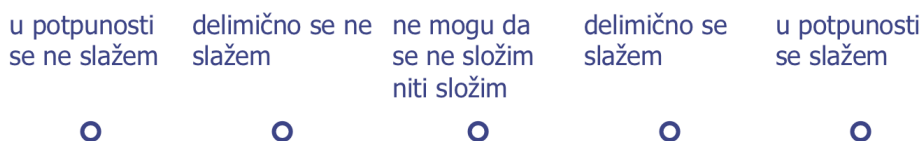
Likert skala je psihometrijski, statistički instrument koji se koristi za merenje preferencija ispitanika ili nivoa slaganja sa izjavama o temama od interesa, u zavisnosti od ciljeva [42]. U kontekstu definisanja zahteva, cilj može biti prikupljanje informacija o aktivnostima ili stavovima ispitanika povezanim sa predmetom istraživanja. Korisnici se izjašnjavaju koristeći uređenu skalu vrednosti (Slika 62a). Najčešće se koristi skala sa pet podeoka, od negativne granične vrednosti kao izraz snažnog neslaganja sa izjavom, preko neutralne vrednosti u sredini, do pozitivne granične vrednosti kao izraza snažnog slaganja sa izjavom. Mogu se koristiti i skale sa 7 podeoka radi veće granularnosti opcija

i rezultata. Skala se može analizirati tako što se svakom podeoku dodeli numerička vrednost tako da negativni ekstrem ima vrednost 1 gde se vrednost za svaki naredni podeok inkrementira za 1.

Skala je jednostavna za analizu i to predstavlja njenu glavnu prednost. Sa druge strane, izjave se moraju pažljivo formulisati sa obzirom na to da se može proceniti nivo slaganja korisnika sa unapred definisanom izjavom. Na ovaj način korisnik ima ograničenu slobodu u davanju odgovora.

Semantička diferencijalna skala je drugačija vrsta psihometrijskog, statističkog instrumenta [43]. Koristi se za vrednovanje specifičnog subjekta od strane ispitanika. Umesto vrednovanja predefinisane izjave, korisnik dopunjuje izjavu (Slika 62b). Granične vrednosti skale su formulisani kao pridevske sintagme koje imaju suprotno značenje, odnosno predstavljaju dijametralne stavove o predmetu pitanja. Skala se formira i analizira na sličan način kao kod likert skale. Sa obzirom na korišćenu jezičku formu (pridevi umesto izraza), ispitanicima može biti lakša i intuitivnija za popunjavanje.

Sajt singidunum.ac.rs je lak za korišćenje.



(a)

Sajt singidunum.ac.rs je



(b)

Slika 62. Uporedni primer (a) likert i (b) semantičke diferencijalne skale.

Upitnik otvorenog tipa od ispitanika zahteva odgovore u slobodnoj formi. Glavna prednost ovog tipa je što ispitanici nisu ograničeni i mogu dati sadržajnije i informativnije odgovore. Istovremeno mogu biti složeniji za analizu zbog količine podataka koje je moguće prikupiti.

Kod dizajna upitnika treba obratiti pažnju na način na koji se formulišu pitanja. Od toga da li i kako korsnici razumeju pitanja u velikoj meri zavisi kvalitet odgovora, odnosno prikupljenih podataka. Ovo se dalje može odraziti i na kvalitet korisničkih zahteva izvedenih analizom podataka.

Pitanja je potrebno definisati tako da budu neutralna i ne navode korisnika na specifičan odgovor. Treba izbegavati takozvana sugerišuća pitanja (eng. *leading questions*). Ova pitanja utiču na korisnika da odgovori na određeni način. Samim tim prikupljeni podaci nisu verodostojni i mogu voditi ka pogrešnim zaključcima (Slika 63).

Pitanje 1: “*Zbog čega ste imali poteškoća prilikom navigacije u aplikaciji ?*”

Pitanje implicira da je korisnik imao problema i navodi na odgovor. Takođe pretpostavlja krivicu korisnika, a ne sistema. Akcenat treba biti na radnjama korisnika, a ne elementima interfejsa.

Pitanje 2: “*Šta je bilo lako ili komplikovano prilikom nalaženja željenih informacija ?*”

Pitanje pomera fokus na temu od interesa, odnosno kretanje i nalaženje sadržaja u aplikaciji. Korisnik može dati odgovor koji želi, a da pri tome ne mora da izbegava neslaganje sa ispitivačem. Na ovaj način, ispitivač se istovremeno fokusira na pravu temu dok korisniku ostavlja potreban prostor za odgovor (ne sugerise odgovor).

Slika 63. Uporedni primer (1) sugerišućeg pitanja i (2) neutralno formulisanog pitanja.

Pored toga, treba izbegavati osetljiva pitanja (eng. *sensitive questions*). Ovo su pitanja koja se odnose na informacije koje mogu zadirati u privatnost korisnika. Korsnici generalno ne žele da otkrivaju neke informacije o sebi (na primer, zdravlje, zarada, emotivni status i druge). Ukoliko se ove informacije prikupljaju od korisnika, pitanja treba formulisati tako da budu zatvorenog tipa sa opcijama koje predstavljaju opsege kao približne vrednosti tako da korisnik ne mora uneti precizniju ili konkretniju informaciju.

6.2.4. Smernice za dizajn upitnika

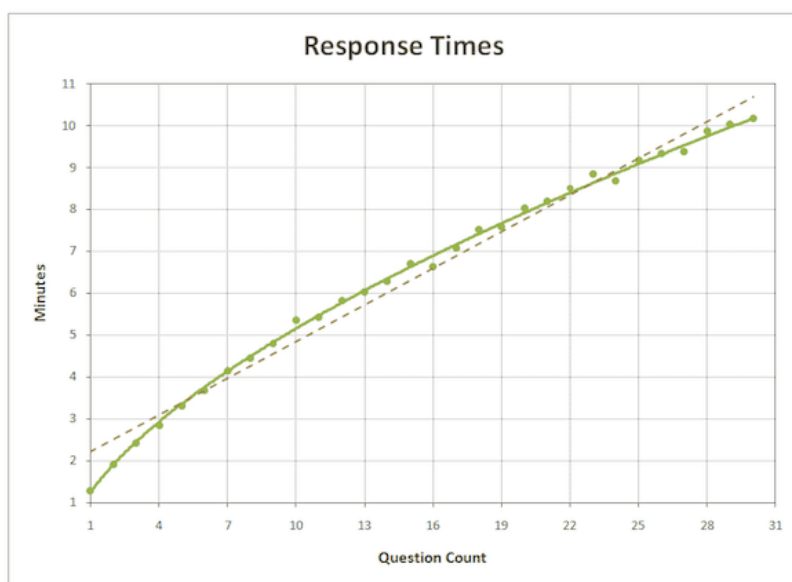
Na osnovu analize upitnika kao tehnike za prikupljanje podataka mogu se formulisati sledeće preporuke:

- Korisnik mora biti informisan o ciljevima upitnika, razlozima prikupljanja podataka, načinima na koji će oni biti korišćeni, i imati sažete instrukcije o načinu popunjavanja.
- Obezbediti anonimnost ispitanika.
- Upitnik početi sa opštim, demografskim pitanjima, nastaviti sa pitanjima vezanim za ciljeve prikupljanja podataka.
- Kombinovati tipove upitnika. Ako se koristi upitnik zatvorenog tipa, poželjno je imati i određeni broj pitanja otvorenog tipa.
- Dužina, odnosno vreme popunjavanja upitnika bi trebalo biti do 10 minuta (Slika 64).
- Pilot-testiranje (eng. *pilot testing*) upitnika sa malim brojem korisnika (N=1-2) radi eventualnih ispravki pre distribucije većoj grupi korisnika.
- Na kraju upitnika se zahvaliti korisnicima na angažovanju.

Question Count	Average Seconds Spent Per Question*	Total Survey Completion Times
1	75	1 min 15 sec
2	40	2 min
3-10	30	2 - 5 min
11-15	25	5 - 7 min
16-25	21	7 - 9 min
26-30	19	9 - 10 min

** Rounded and grouped for illustrative purposes*

(a)



(b)

Slika 64. Dužina upitnika - (a) korisnici posvećuju više vremena pojedinačnim pitanjima kada je upitnik kraći; i (b) procenat nedovršenih upitnika se značajnije povećava za upitnike koji traju duže od 7-8 minuta.

Izvor: SurveyMonkey,

https://www.surveymonkey.com/curiosity/survey_completion_times/.

6.2.5. Intervju i fokus grupa

Intervju se može definisati kao konverzacija sa krajnjim korisnikom radi prikupljanja specifičnih podataka od interesa. Postoje četiri osnovna tipa intervjuja [44]:

- Otvoreni ili nestruktuirani intervju (eng. *unstructured interview*),
- Struktuirani intervju (eng. *structured interview*),
- Polu-struktuirani intervju (eng. *semi-structured interview*),
- Grupni intervju ili fokus grupa (eng. *focus group*).

Prva tri tipa se razlikuju na osnovu nivoa kontrole koji ispitivač ima u toku intervjuja, odnosno da li su pitanja i protokol unapred pripremljeni i u kojoj meri su fleksibilni u toku intervjuja. Dok su ovi tipovi individualni intervjui, četvrti tip podrazumeva intervjuisanje manje grupe ispitanika. Intervjui su pogodni za ispitivanje specifičnih tema ili problema. Istovremeno zahtevaju prisustvo korisnika i vremenski su zahtevniji.

Izbor tipa zavisi od ciljeva intervjuja, i faze razvoja u opštem slučaju. Ukoliko je cilj stav korisnika prema ideji za dizajn, nestruktuirani upitnik može biti pogodniji. Ako je cilj povratna informacija o elementu dizajna (na primer, raspored elemenata interfejsa), tada je struktuirani upitnik prikladniji zbog toga što su ciljevi specifičniji.

Nestruktuirani intervju ne predviđa unapred definisana pitanja i njihov redosled. Intervju je istraživačke prirode (eng. *exploratory*) i podseća na razgovor o specifičnoj temi.

Ispitivač u toku razgovora formuliše i postavlja pitanja, i tok intervjuja je fleksibilan i zavisi od sadržaja odgovora. U praksi se obično kreće sa opštijim pitanjima u vezi sa temom, a zatim pitanja postaju specifičnija u vezi sa produbljivanjem teme, odnosno njenim konkretnim elementima.

Bez obzira što pitanja i tok nisu prethodno definisani, preporuka je imati pripremljene teme intervjuja. Kod otvorenog intervjuja važno je uspostaviti ravnotežu između razgovora o postojećim temama, i novim, relevantnim temama koje se mogu pojaviti u toku razgovora.

Prednost otvorenog intervjuja jesu podaci koji mogu omogućiti otkrivanje i razumevanje relevantnih tema, ili tema koje u početku nisu bile poznate a važne su za dizajn sistema. Istovremeno mogu zahtevati značajno vremensko angažovanje u toku intervjuja i analize podataka. Obično generišu velike količine nestruktuiranih, tekstualnih podataka koji su složeni za analizu. Sa stanovišta metodologije istraživanja, repliciranje ovog intervjuja može biti problem zbog toga što razgovor sa svakim od ispitanika može biti drugačiji.

U kontekstu dizajna interaktivnog proizvoda, otvoreni intervju je pogodan za razumevanje aktivnosti ili procedura korisnika, stavova prema ovim aktivnostima, i njihovih očekivanja i novih potreba.

Strukturirani intervju se odvija prema unapred definisanim pitanjima i redosledu. Intervju je identičan za svakog ispitanika. Pitanja moraju biti kratka i jasno formulisana. Pitanja mogu biti i zatvorenog tipa i sadržati skup opcija kao moguće odgovore. Ovaj tip intervju je pogodan za konkretne i specifične ciljeve i aktivnosti (Slika 65).

Otvoreno pitanje: Koje su prednosti korišćenja dodirnog korisničkog interfejsa ?

(a)

Zatvoreno pitanje: Koliko često posećujete ovaj Veb sajt: svakog dana, jednom u nedelji, jednom u mesecu, ili ređe od jednom mesečno ?

(b)

Slika 65. Intervju - primeri pitanja (a) otvorenog i (b) zatvorenog tipa.

Polu-strukturirani intervju kombinuje dva prethodna tipa [20]. Ispitivač ima unapred pripremljen plan intervju tako da sve teme budu diskutovane sa svim ispitanicima. Intervju obično počinje planiranim pitanjima i zatim postavlja nova pitanja u zavisnosti od toka razgovora, odnosno ukoliko se pojave nove, relevantne informacije. Ovaj tip ublažava nedostatke otvorenog i zatvorenog intervju, i zbog toga se najčešće koristi. Sa jedne strane, omogućava ispitivaču da istraži teme koje se mogu pojaviti, a nisu predviđene planom intervju. Sa druge strane, ispitanicima pruža veću slobodu u razgovoru i izražavanju. Slika 66 ilustruje tipičan tok intervju sa aspekta angažovanja korisnika.



Slika 66. Tipičan tok intervjua sa aspekta angažovanja korisnika.

Fokus grupa je intervjui sa grupom korisnika. Često se koristi u marketingu i društvenim naukama. Ispitivač kao moderator vodi diskusiju za tri do pet ispitanika. Ispitanici se pažljivo biraju tako da predstavljaju reprezentativni uzorak ciljne grupe korisnika. Na primer, prilikom dizajna sajta univerziteta, mogu postojati tri fokus grupe - studenti, administrativno osoblje i predavači zbog toga što će tri grupe koristiti sajt na različite načine i zbog toga ima različite zahteve. Kod fokus grupe treba voditi računa da svi učesnici podjednako učestvuju u diskusiji i izbegavati dominaciju jedne člana grupe.

6.2.6. Smernice za intervjui

Na osnovu analize intervjua kao tehnike za prikupljanje podataka mogu se formulisati sledeće preporuke:

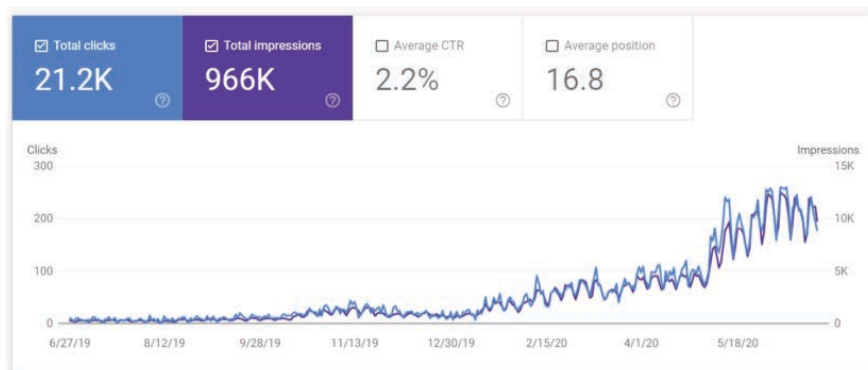
- Pre početka intervjua potrebno je obezbediti prostoriju, opremu (ili softver) za snimanje, pitanja i obrazac pristanka (eng. *informed consent form*).
 - Obrazac pristanka u pisanom obliku ispitanika informiše o ciljevima intervjua, razlozima prikupljanja podataka, načinima na koji će oni biti korišćeni, i očuvanju anonimnosti i privatnosti podataka o korisniku.

- Na početku intervjua, ispitanik se predstavlja, objašnjava ciljeve intervjua, obavještava korisnika i traži pristanak za snimanje intervjua, i pita korisnika za potvrdu (ili potpis) obrasca pristanka.
- Pitanja rasporediti u skladu sa tipičnom tokom intervjua sa aspekta angažovanja ispitanika (Slika 66).
- Pitanja formulisati tako da budu neutralna.
 - Izbegavati dugačke rečenice ili pitanja koja se odnose na veći broj tema (jedno pitanje, jedna tema).
 - Izbegavati termine koje ispitanik možda neće razumeti.
 - Izbegavati sugerišuća pitanja.
- Ispitivač se mora ponašati neutralno.
 - Ne treba se niti slagati niti neslagati sa ispitanikom (verbalno i neverbalno).
- Po potrebi, koristiti potpitanja za dodatne informacije,
- Na kraju se zahvaliti se ispitaniku i zaustaviti snimanje.

6.2.7. Logovanje podataka

Ova tehnika se koristi u slučaju da postoji sistem u probnoj ili operativnoj upotrebi. Podrazumeva implementaciju i izvršavanje programa koji snima i čuva podatke vezane za aktivnosti korisnika unutar aplikacije radi kasnije analize. Podaci mogu biti tekstualni (kao što su komentari korisnika) i/ili numerički. U većoj meri se prikupljaju numerički podaci poput identifikatora selektovanih elemenata interfejsa i stranica, brojeva grešaka, vremena izvršenja zadataka na interfejsu, vremena provedenih na specifičnim stranicama, informacija o tome koje delove stranice korisnik posmatra i drugih.

Postoje i gotovi Veb alati koji se mogu koristiti, kao što je Gugl Analitike [45] (Slika 67).



Slika 67. Primer Gugl Analitike alata za prikupljanje, analizu i vizuelizaciju podataka o aktivnostima korisnika bloga. Izvor: <http://datahacker.rs/>.

6.2.8. Smernice za tehnike prikupljanja podataka o korisnicima

Tabela 1 prikazuje uporedni, sumarni pregled tehnika za prikupljanje podataka u odnosu na kriterijume u zaglavlju.

Tabela 1. Uporedni pregled tehnika za prikupljanje podataka pri definisanju zahteva korisnika.

Tehnika	Namena	Podaci	Prednosti	Mane
<i>Analiza dokumentacije</i>	Upoznavanje pravila, procedura i postupaka	Kvantitativni	Ne zahteva angažovanje krajnjih korisnika	Praktična rutina se može razlikovati od dokumentovanih procedura
<i>Posmatranje</i>	Razumevanje konteksta aktivnosti korisnika	Kvalitativni	Neposredno posmatranje može dati podatke koje druge tehnike ne mogu prikupiti	Vremenski zahtevno i generiše veliku količinu podataka
<i>Upitnik</i>	Odgovori na specifična pitanja	Kvantitativni i kvalitativni	Veliki broj učesnika sa malo ulaganja	Dizajn je kritičan. Nedovoljan odziv. Neodgovarajući odgovori
<i>Intervju</i>	Istraživanje tema	Pretežno kvalitativni, manje kvantitativni	Ispitivač vodi ispitnika kroz proces. Direktni kontakt između dizajnera i korisnika	Vremenski zahtevan. Okruženje može uticati na ispitanike
<i>Fokus grupa</i>	Različita posmatranja jedne teme u trenutku	Pretežno kvalitativni, manje kvantitativni	Uočavanje predmeta slaganja ili konflikata. Direktni kontakt između dizajnera i korisnika	Mogućnost pojave dominantnih korisnika u grupi
<i>Logovanje podataka</i>	Neposredan uvid u probleme	Više kvantitativni, manje kvalitativni	Jednostavna i efikasna realizacija	Vreme potrebno za prikupljanje podataka. Nedovoljno angažovanje korisnika

Tehnike prikupljanja podataka se mogu kombinovati. Prednost je što se može prikupiti više podataka koji opisuju teme od interesa sa različitih aspekata koji se dopunjuju. Samim tim se mogu izvesti precizniji i verodostojniji rezultati analizom ovih podataka. Sa druge strane, korišćenje više tehnika zahteva i više vremena i angažovanja.

U opštem slučaju, na izbor pogodne tehnike utiču sledeći faktori:

- Cilj prikupljanja podataka koji određuju i vrste podataka koju je potrebno prikupiti.
- Ciljna grupa korisnika, odnosno osobine i raspoloživost korisnika za prikupljanje podataka.
- Osobine i zahtevi tehnike, kao što su da li zahteva posebnu opremu ili posebne veštine ispitivača.
- Raspoloživi resursi u pogledu vremena i ljudskih resursa.

6.3. Obrada prikupljenih podataka

Tehnika obrade je pre svega određena vrstom podataka koja se prikuplja na osnovu definisanih ciljeva. Sa aspekta obrade i interpretacije podataka, podaci se mogu svrstati u:

- Kvalitativni podaci - izraženi u opisnoj (eng. *descriptive*) formi kao tekst, slike ili video sadržaj.
- Kvantitativni podaci - izraženi u numeričkoj formi, odnosno koji se mogu predstaviti i analizirati kao brojevi.

Primena specifičnih tehnika kreira kvalitativne i kvantitativne podatke (Tabela 2). Iako se tipovi podataka pojavljuju u različitim udelima, važno je imati na umu da svaka tehnika generiše obe vrste podataka koji se mogu analizirati.

Tabela 2. Primeri kreiranih podataka u najzastupljenijim tehnikama prikupljanja podataka.

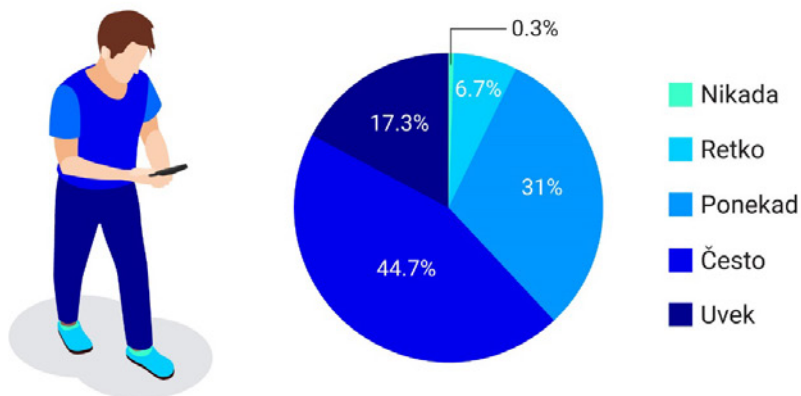
Tehnika	Format podataka	Kvalitativni podaci	Kvantitativni podaci
<i>Intervju (fokus grupa)</i>	Audio i/ili video zapisi. Pisane beleške	Odgovori na otvorena pitanja. Audio/video snimci. Fotografije	Odgovori na zatvorena pitanja
<i>Upitnik</i>	Tekstualni odgovori	Odgovori na otvorena pitanja	Odgovori na zatvorena pitanja
<i>Logovanje podataka</i>	Tekstualni zapisi. Numerički zapisi	Tekstualni podaci poput komentara i tekstualnog unosa	Numerički podaci poput vremena izvršenja zadatka, broja klikova, i drugi

Kvantitativna analiza podrazumeva korišćenje numeričkih metode obrade podataka kako bi rezultat bio izražen kao brojčana vrednost. Na primer kao stepen, veličina ili količina, u zavisnosti od prirode informacije.

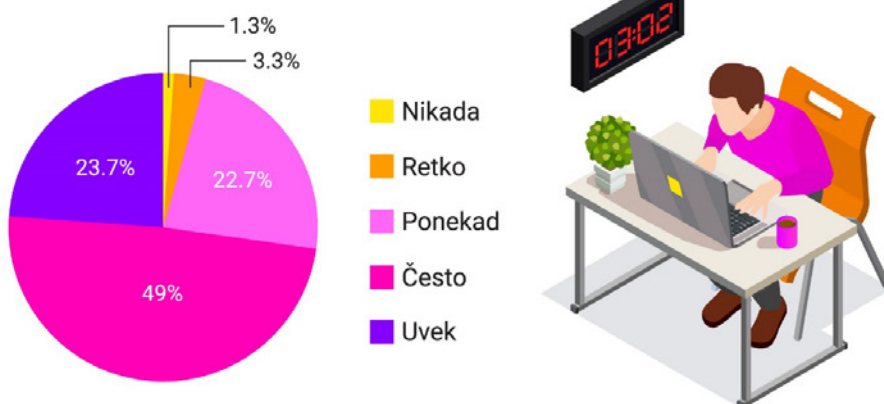
Ovde spadaju različite statističke metode, od jednostavnijih do složenijih. Često se koriste procenti i srednje vrednosti. Procenti su pogodni za prikazivanje odnosa i poređenje između većeg broja skupova podataka (Slika 68). Srednje vrednosti se koriste za kvantifikovanje specifičnih pojava ili osobina grupa podataka (Slika 69). Na primer, *mean* kao zbir vrednosti podeljen sa brojem vrednosti, *median* kao središnja vrednost u nizu rangiranih vrednosti, *mode* kao vrednost koja sa najčešće pojavljuje, i *standardna devijacija* vrednosti iz skupa u odnosu na *mean* vrednost.

Mogu se koristiti i softverski alati kao što je *Microsoft Excel* za jednostavnije proračune, i programi za naprednije statističke analize kao što su *R* i *SPSS*.

Koliko često “odlutate” u korišćenju društvene mreže nakon što ispunite primarni cilj njenog korišćenja?



Koliko često mislite da bespotrebno provodite vreme na društvenim mrežama?



Slika 68. Kvantitativna analiza - procenti su pogodni za ilustraciju udela i međusobnog odnosa podskupova podataka u okviru skupa (N=300 ispitanika). Preuzeto iz [46].

Table 11. Mean ratings of subjective ease of use and satisfactions for features (the scale range is 1-5).

Feature	Mean rating of ease of use	Mean rating of satisfaction
Profile	4.6	4.3
Groups	3.5	3
Stories	3.7	3.5
Good practices	3.1	2.7
Public benefits	3.1	2.7
Commonplace	3.9	3.8
Wallet	4.5	3.9
Search and navigation	4	4.1
Communication	3.1	3.5
Website (overall)	3.3	3.6

Slika 69. Kvantitativna analiza - srednje vrednosti (kolone) se mogu koristiti za kvantifikovanje osobina grupa podataka. Svaka grupa (red) opisuje deo korisničkog interfejsa Veb sajta.

Kvalitativna analiza se svodi na nalaženje objašnjenja u podacima na taj način što se uočavaju teme ili obrasci koji se ponavljaju. Da bi se ovaj proces omogućio, intervjui se snimaju a nakon toga se vrši transkripcija audio ili video zapisa. Jedna od najčešće korišćenih tehnika jeste tematska analiza (eng. *thematic analysis*) [47]. Kvalitativna analiza je vremenski zahtevna. Zbog toga postoje alati koji automatizuju deo procesa, kao što su *Atlas.ti* i *NVIVO*.

Važno je imati na umu da su obe vrste analiza, i kvalitativna i kvantitativna, podjednako značajne zbog toga što nam daju različita, ali komplementarna saznanja. Konkretno, kvantitativna analiza otkriva pojave, odnosno daje odgovor na pitanje ŠTA se dešava. Sa druge strane, kvalitativna analiza nam otkriva razloge nastanka pojave, odnosno daje odgovor na pitanje ZBOG ČEGA. Na primer, ukoliko korisnici u manjoj meri koriste odgovarajuću funkciju aplikacije, to možemo otkriti analizom logovanih podataka (kvantitativna analiza). Da bismo saznali razlog pojave moramo razgovarati sa korisnicima putem intervjua ili upitnika (kvalitativna analiza). Dok kombinovanje različitih analiza (i tehnika prikupljanja podataka) iziskuje dodatno angažovanje, može dati bolje rezultate.

7. Prototipiziranje i dizajn

Aktivnosti vezane za konkretan dizajn interaktivnog proizvoda počinju nakon uspostavljanja korisničkih zahteva. Kako bi korisnici što ranije mogli da procene dizajn interaktivnog proizvoda, ideje se mogu prototipizirati, odnosno realizovati i konkretizovati na jednostavniji i efikasan način tako da podsećaju na krajnji proizvod. Celokupan proces dizajna se odvija u većem broju iteracija. U ranijim iteracijama, prototipi mogu biti nacrtani na papiru ili se postojeće komponente mogu kombinovati radi testiranja sa korisnicima. Kako proces dizajna napreduje, prototipi postaju sve detaljniji i kompaktniji tako da u sve većoj meri liče na krajnji proizvod.

Šire posmatrano, proces dizajna može početi bez postojećeg sistema, ili predstavljati modifikaciju postojećeg rešenja. U oba slučaja, prototipiziranje je važno. Čak i u smanjenom obimu, odnosno kada se dodaju novi ili menjaju postojeći elementi sistema u upotrebi.

7.1. Prototipi - definicija i motivacija

Poznato je da korisnici mogu imati poteškoća da definišu kakav proizvod žele [12]. Međutim, kada vide ili koriste konkretnu reprezentaciju ili sliku budućeg sistema, oni mogu lakše i bolje formulisati šta žele, i šta ne žele od budućeg sistema. Iz ovoga proističe i motivacija za prototipiziranje kao važnog elementa u razvoju interaktivnih tehnologija.

Prototip predstavlja reprezentaciju ili manifestaciju dizajna proizvoda tako da korisnici mogu interagovati sa njom i dati povratnu informaciju. U skladu sa ovom definicijom, prototip može biti skica ekrana na papiru, kolekcija komponenti vizuelnog dizajna, digitalna slika, video simulacija, softverska ili harverska komponenta, ili trodimenzionalni model računara. Ono što je zajedničko za prethodne primere je što korisniku mogu obezbediti predstavu o tome kako bi budući sistem mogao da izgleda i da se ponaša.

Prototipi su zaslužni za pojavu mnogih savremenih tehnologija koje koristimo. Na primer, prvi uređaj, koji se smatra i pretečom savremenih mobilnih tehnologija, je *PalmPilot*. Uređaj je nastao tako što je njegov tvorac, Džef Hokin (eng. *Jeff Hawkin*), u komadu drveta koji se može držati na dlanu isklesao i nacrtao interfejs uređaja (eng. *handheld device*). On je sa sobom nosio prototip i pretvarao se da koristi interfejs samo da bi stekao utisak o tome kako bi izgledalo imati i koristiti takav uređaj u svakodnevnom životu [48].



Slika 70. *PalmPilot* uređaj (levo) i prototip od drveta (desno). Preuzeto i adaptirano iz [48].

Sledeći, skoriji primer jeste trodoimenzionalna (3D) štampa. Napredak tehnologija 3D štampača, uz redukovanje cena i povećanje pristupačnosti, doveo je do značajnije upotrebe u dizajnu. Tako je, na primer, moguće odštampati 3D model iz programa kao 3D predmet u stvarnosti (Slika 71).



(a)



(b)

Slika 71. 3D štampa se koristi za prototipiziranje (a) protetičkih uređaja, i (b) modela uređaja miša. Izvori: (a) <https://www.seeedstudio.com/blog/> i (b) <https://www.3dhubs.com/> (15.11.2020).

Prototipi su korisni za diskutovanje ili evaluaciju ideja sa krajnjim korisnicima. Na ovaj način se može dobiti rana povratna informacija od korisnika pre većih ulaganja u realizaciju gotovog rešenja koja nose rizike sa aspekta upotrebljivosti. Prototipi se brže i lakše realizuju, mogu se rano testirati i dobiti informacije o lošim i dobrim stranama dizajna. Može se reći da služe kao sredstvo

komunikacije između dizajnera i korisnika - korisnik zna šta dizajner misli tek kada vidi ideju.

Slično prethodnom, predstavljaju sredstvo komunikacije unutar razvojnog tima. Na primer, eksperimentisanje sa alternativama dizajna. Ovo znači da se može sa kreirati i uporediti veći broj jeftinih realizacija jedne ideje. Pored toga su lakši za promene i problemi u dizajnu se mogu lakše otkloniti ako se ranije otkriju. Konkretniji praktični razlozi za prototipiziranje uključuju procenu tehničke izvodljivosti ideje, pojašnjenje korisničkih zahteva koji nisu definisani dovoljno precizno, testiranje specifičnih elemenata dizajna sa korisnicima, i uvid da li je dizajn korisničkog interfejsa kompatibilan sa drugim delovima sistema.

Svrha prototipa određuje vrstu prototipa koja će biti kreirana. Na primer, ako se želi pojasniti kako bi korisnici mogli obavljati zadatke na interfejsu i da li im predloženi dizajn to omogućava, može se kreirati papirni prototip korisničkog interfejsa (Slika 72a). Papirni prototip pokazuje raspoložive funkcije i vidljive opcije koje ih pokreću, raspored vizuelnih elemenata i njihove nazive ili simbole, i dimenzije i oblik uređaja na kojem bi budući sistem trebalo da se izvršava. Nijedan element nije aktivan, ali korisniku mogu dati neophodne informacije o tome da li dizajn odgovara njegovim potrebama. Drugim rečima, ovakav prototip je dovoljan za ispitivanje scenarija korišćenja i uvid, na primer, da li su nazivi ili simboli dugmadi pogodni i da li su pokrivene potrebne funkcionalnosti sistema.

U toku razvoju sistema pametne ishrane (eng. *Smart Food Nutrition Scale, SITU*) (Slika 72), varijante prototipa su kreirane u različitim fazama i ciklusima, od jednostavnijih ka složenijim.



(a)



(b)



(c)

Slika 72. Prototipi u razvoju *SITU* aplikacije: (a) nacrtana skica sistema na papiru, (b) štampani vizuelni dizajn sistema, i (c) interaktivni digitalni prototip.

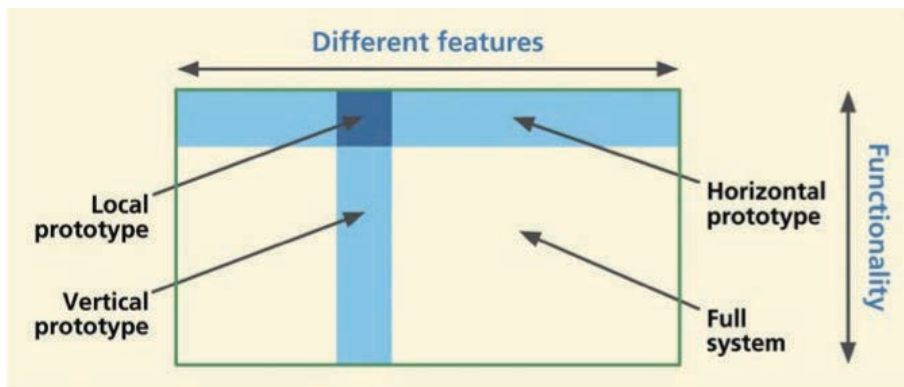
Izvor: <https://www.kickstarter.com/projects/situ/situ-smart-food-nutrition-scale>.

Opšteprihvaćeni kriterijum za klasifikaciju prototipa jeste verodostojnost (eng. *fidelity*). Verodostojnost se može definisati kao nivo sličnosti prototipa sa krajnjim proizvodom [49]. Na ovaj način prototipi mogu biti niske verodostojnosti (eng. *low-fidelity prototype*) i visoke verodostojnosti (eng. *high-fidelity prototype*). Primer prototipa niske verodostojnosti jeste papirna skica interfejsa (Slika 72a), dok primer prototipa visoke verodostojnosti može biti interaktivna verzija korisničkog interfejsa bez ili sa simuliranom pozadinskom logikom (Slika 72c).

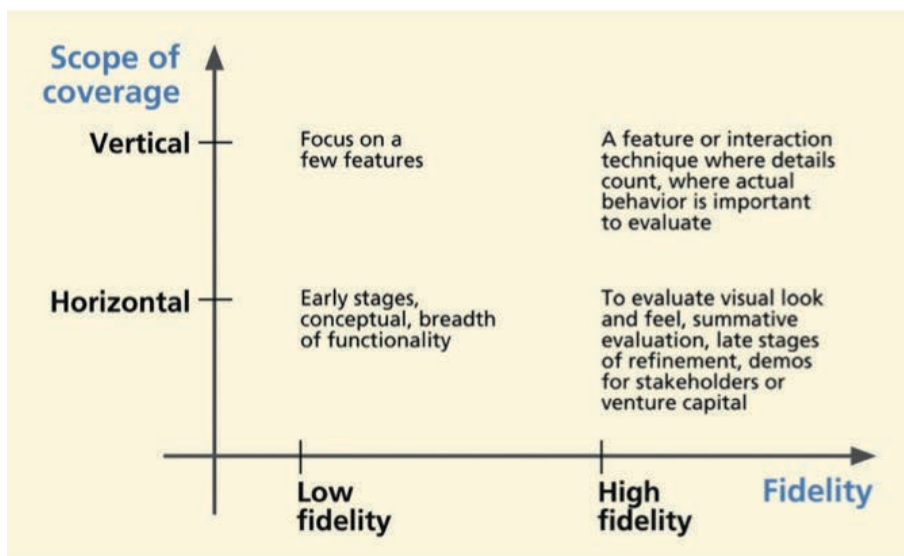
Kao dodatak, ponekada se koristi i pojam, odnosno kategorija prototipa srednjeg nivoa verodostojnosti (eng. *medium-fidelity prototype*). Prototipi ove kategorije obično realizuju detaljniji prikaz i ponašanje u odnosu na prototip niske verodostojnosti, dok su u istim aspektima oskudniji u odnosu na prototip visoke verodostojnosti. Dakle, interpretiraju se relativno u odnosu na dve granične kategorije i koriste se po potrebi, u zavisnosti od toka celokupnog procesa dizajna.

Verodostojnost je složen koncept i može se posmatrati šire, u kontekstu celokupnog sistema koji se razvija. Tako se može raščlaniti na dve dimenzije - verodostojnost širine (eng. *breadth*) i verodostojnost dubine (eng. *depth*). Širina označava udeo funkcija sistema koje prototip realizuje. Dubina označava stepen realizacije specifične funkcije. Slika 73a ilustruje značenje dve dimenzije. Horizontalni prototip (eng. *horizontal prototype*) je širok i plitak. Drugim rečima, korisnički interfejs sistema sa svim funkcijama (eng. *frontend*) bez implementirane pozadinske logike (eng. *backend*). Vertikalni prototip (eng. *vertical prototype*) je uzak i dubok, odnosno korisnički interfejs sa jednom ili više funkcija koje su implementirane i imaju pozadinsku logiku.

Izbor dimenzije zavisi od ciljeva procesa dizajna. U razvoju korisničkog interfejsa, akcenat je na horizontalnim prototipima kako bi se unapredila upotrebljivost kao glavni cilj. U razvoju aplikacije realizacije funkcija mogu biti kritične. Na primer, kada je potrebno utvrditi da li se funkcionalni zahtevi mogu realizovati. U tom slučaju akcenat je na vertikalnom prototipima za verifikaciju funkcija sistema (ovde je glavni cilj korisnost). Specijalan slučaj je lokalni prototip (eng. *local prototype*) koji realizuje jednu funkciju po dubini. Ova vrsta prototipa se koristi kod modifikacije ili proširenja postojećih aplikacija kako bi se videli izgled i ponašanje dela interfejsa za datu funkcionalnost.



(a)



(b)

Slika 73. Prototipiziranje - (a) klasifikacija i (b) analiza prototipa na osnovu verodostojnosti kao složenog kriterijuma koji se može posmatrati po širini i dubini sistema koji se razvija. Preuzeto iz [49].

Slika 73b ilustruje načine na koje se mogu kombinovati dimenzije verodostojnosti kao tipične scenarije ili situacije u praksi. Horizontalni prototip niske verodostojnosti se koristi rano u dizajnu kao koncept interfejsa koji ispunjava potrebe korisnika. Kako dizajn napreduje ovaj prototip prerasta u prototip visoke verodostojnosti koji u velikoj meri podseća na krajni proizvod i može se koristiti za završna testiranja. Vertikalni prototip niske verodostojnosti

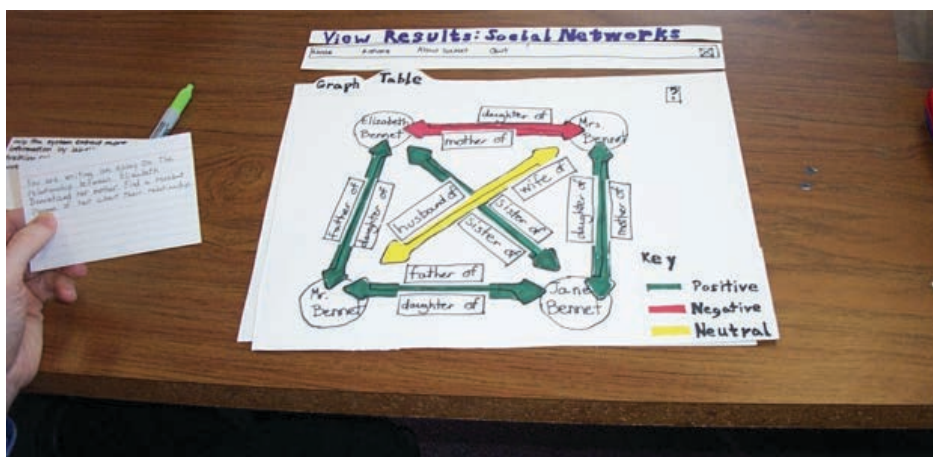
se fokusira na izgled dela funkcija sistema, dok visoka verodostojnost označava kompletnu implementaciju funkcija gde je akcenat na ponašanju sistema.

U nastavku će biti detaljnije opisani prototipi niske i visoke verodostojnosti.

7.2. Prototipi niske verodostojnosti

Prototip niske verodostojnosti predstavlja grubu skicu ili nacrt planiranog korisničkog interfejsa. Obično se koriste materijali poput papira kako bi se predstavili izgled i funkcije sistema bez demonstracije njihovog izvršenja. Dizajner crta interfejs kako bi ga pokazao korisniku. Interfejs može biti nacrtan ručno i direktno na papiru, ili korišćenjem programa koji omogućavaju crtanje čiji se rezultati štampaju. Osnovna prednost je to što su jednostavni, brzi i jeftini za kreiranje. Ne iziskuju značajnije vreme niti posebne veštine za izradu. Važna posledica je da su laki za promeniti tako da omogućavaju i stimulišu ispitivanje različitih realizacija ideje kao alternativa (varijanti) dizajna. Ovo je ključna aktivnost u ranim fazama dizajna, prilikom smišljanja i razrade koncepta interfejsa. Prototipi za testiranje ideja moraju biti fleksibilni i laki za promene. Primarna svrha ovih prototipa jeste istraživanje i ispitivanje ideja na početku procesa razvoja interaktivnog proizvoda.

Papirni prototipi se najčešće koriste u svrhu ranog prototipiziranja. Predstavljaju fizičku ili materijalnu skicu koncepta interfejsa. Sam dizajn je stvar kreativnosti autora. U praksi se obično koriste krupniji delovi papira za osnovne prikaze interfejsa kao što su prozori i paneli. Manji delovi se koriste za dinamičke elemente koji se pojavljuju preko osnovnog prikaza, kao što su meniji, forme za unos, i dijalozi sa porukama (Slika 74a). Prilikom pokazivanja prototipa korisnicima dizajneri simuliraju ponašanje interfejsa pomeranjem ili premeštanjem delova interfejsa, i rečima opisuju akcije ili funkcije koje se izvršavaju.



(a)

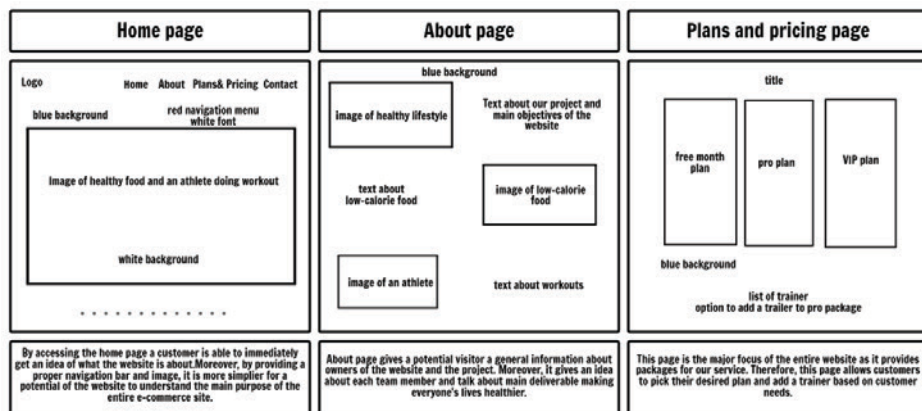


(b)

Slika 74. Papirni prototipi niske verodostojnosti se koriste na početku procesa razvoja - (a) skica interfejsa aplikacije društvenih mreža, i (b) nacrt interfejsa mobilne aplikacije za promovisanje fizičke aktivnosti.

Pored skiciranja izgleda korisničkog interfejsa, papirni prototipi se koriste i za opis same operacije ili interakcije korisnika sa interfejsom. Tehnika je poznata pod nazivom narativi (eng. *storyboarding*). Narativ (eng. *storyboard*) se sastoji od sekvence ili skupa skica (eng. *sketch*) koje pokazuju kako se koristi može kretati kroz aplikaciju, i koje funkcije ima na raspolaganju u specifičnim koracima ilustrovanim delovima interfejsa (Slika 75). Konkretna zadatak je

često navigacija, međutim može biti i drugi. Iako kreira skice niske verodostojnosti, tehnika koja povezuje skice u smislenu celinu daje potpuniju i pregledniju predstavu o tome kako bi interfejs mogao da izgleda i da se ponaša. Samim tim se može dobiti i upotrebljivija povratna informacija od krajnjih korisnika. Ova tehnika je naročito korisna u dizajnu Veb sajtova složenije strukture kod kojih navigacija predstavlja jedan od ključnih problema upotrebljivosti [50].



Slika 75. Narativ koji ilustruje povezane elemente aplikacije kroz koje se korisnik može kretati.

Izvor: <https://www.storyboardthat.com/storyboards/salome5428/salome>.

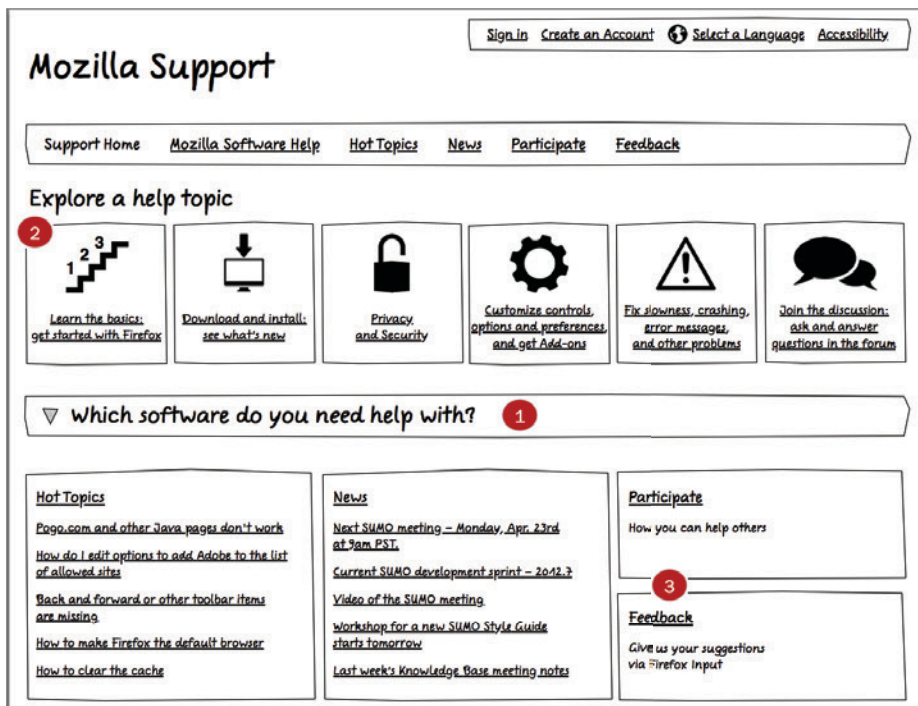
Opšte posmatrano, testiranje papirnog prototipa nam može otkriti sledeće informacije:

- Konceptualni model interfejsa - da li korisnik razume namenu i principe interfejsa.
- Funkcionalnost - da li prototip sadrži funkcije potrebne korisniku.
- Navigacija - da li korisnik zna kako da se kreće kroz interfejs.
- Tok zadatka - da li korisnik razume korake ili faze izvršenja zadatka.
- Terminologija - da li korisnik razume jezik interfejsa.
- Sadržaj - da li su sadržane informacije potrebne korisniku.

Osnovne prednosti upotrebe papirnih prototipa obuhvataju:

- Rano otkrivanje konceptualnih problema dizajna. Zbog toga što akcenat nije na detaljima (na primer, font, boja, ili ravnjanje), korisnici mogu dati kreativnije sugestije (Slika 76).
- Brzo se kreiraju.

- Lakši su za promenu, čak i u toku testiranja sa korisnicima.
- Ne zahtevaju posebne veštine poput grafičkog dizajna, modelovanja ili programiranja.



Slika 76. Papirni prototipi su značajni za testiranje ideja i koncepata interfejsa u ranim fazama procesa razvoja.

Izvor: Test Paper Prototypes to Save Time and Money: The Mozilla Case Study: <https://www.nngroup.com/articles/mozilla-paper-prototype/>.

Na osnovu prethodnih razmatranja mogu se definisati preporuke za izradu papirnih prototipa:

- Poželjno je da budu većih dimenzija kako bi testiranje bilo olakšano.
 - Korišćenje prstiju samo po sebi zahteva veće dimenzije papirnog interfejsa.
- Jednostavan vizuelni dizajn.
 - Ograničen skup boja (jedna ili nekoliko) i jednostavan izgled elemenata interfejsa (bez detalja).
 - Složena povratna informacija može biti i izgovorena.
- Čuvanje delova u slučaju modifikacije i ponovnog korišćenja.

- Fascikla za krupnije delove (na primer ekrani) i koverta za sitnije delove (na primer ponavljajući elementi poput trake sa opcijama, menija i dijaloga za poruke).

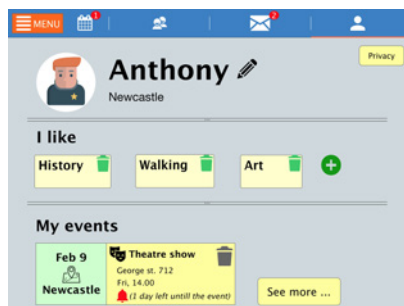
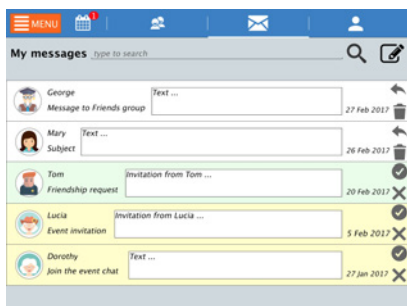
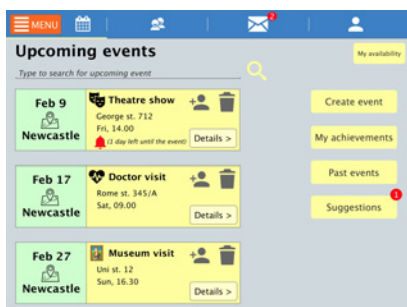
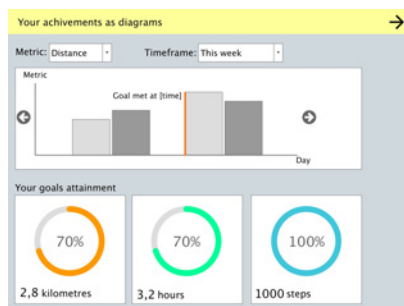
7.3. Prototipi visoke verodostojnosti

Ova vrsta prototipa izgledom podseća na krajnji proizvod i realizuje više funkcionalnosti u odnosu na prethodni. Kreiraju se u kasnijim i završnim fazama procesa razvoja interaktivnog proizvoda. Služe za testiranje detalja izgleda i funkcionisanja proizvoda, i za njegovu promociju i prodaju. Mogu biti kreirani i modifikacijom i kombinovanjem postojećih komponenti - hardverskih i softverskih. U izradi robitičkih uređaja ovakav pristup je poznat pod nazivom tinkering (eng. *tinkering*) [51]. Najčešće su realizovani kao digitalni prototipi koji predstavljaju interaktivnu softversku simulaciju interfejsa. U kontekstu dizajna korisničkog interfejsa ovi prototipi su tipično horizontalni. Kao takvi poseduju visoku verodostojnost izgleda i interaktivnog ponašanja, i nisku verodostojnost dubine bez potpune i prave implementacije funkcija pozadinske logike.

Za kreiranje prototipa visoke verodostojnosti se mogu koristiti alati za prototipiziranje (eng. *prototyping tools*), i programski jezici kao biblioteke i tehnologije za izradu grafičkih korisničkih interfejsa kao što su HTML, CSS i JavaScript jezici, React biblioteka, ili AngularJS okvir (eng. *framework*). Slika 77 daje primer interaktivnog prototipa korisničkog interfejsa kreiran korišćenjem alata za prototipiziranje.

Alati za prototipiziranje ispoljavaju određene prednosti u odnosu na programiranje prototipa interfejsa. Pre svega, jednostavniji su za učenje i brže se koriste, odnosno prototipi se brže kreiraju. Pored toga, kreirani prototipi su lakši za promene nakon testiranja sa korisnicima. Na kraju pružaju veću slobodu i kreativnost u dizajnu interfejsa. Mnoge biblioteke programskih jezika imaju karakteristične i predefinisane šablone dizajna interfejsa u pogledu izgleda, ponašanja i rasporeda elemenata. Na ovaj način mogu ograničiti slobodu i kreativnost u dizajnu. Analizom izgleda i ponašanja korisničkih interfejsa mogu uočiti ovi obrasci i zaključiti u kom jeziku, biblioteci ili okviru su implementirani.

Ovi alati tipično kreiraju interaktivnu reprezentaciju korisničkog interfejsa kao skup povezanih prikaza ili stranica. Skup je organizovan hijerarhijski u obliku stabla ili grafa. Stranice su povezane vezama koje omogućavaju navigaciju kroz prototip. Za izradu prototipa se koriste specijalizovani alati kao što su *AdobeXD*, *Mockplus*, *Balsamiq* i *MarvelApp*. Ovi alati se koriste i za kreiranje prototipa niske verodostojnosti (eng. *wireframe*).

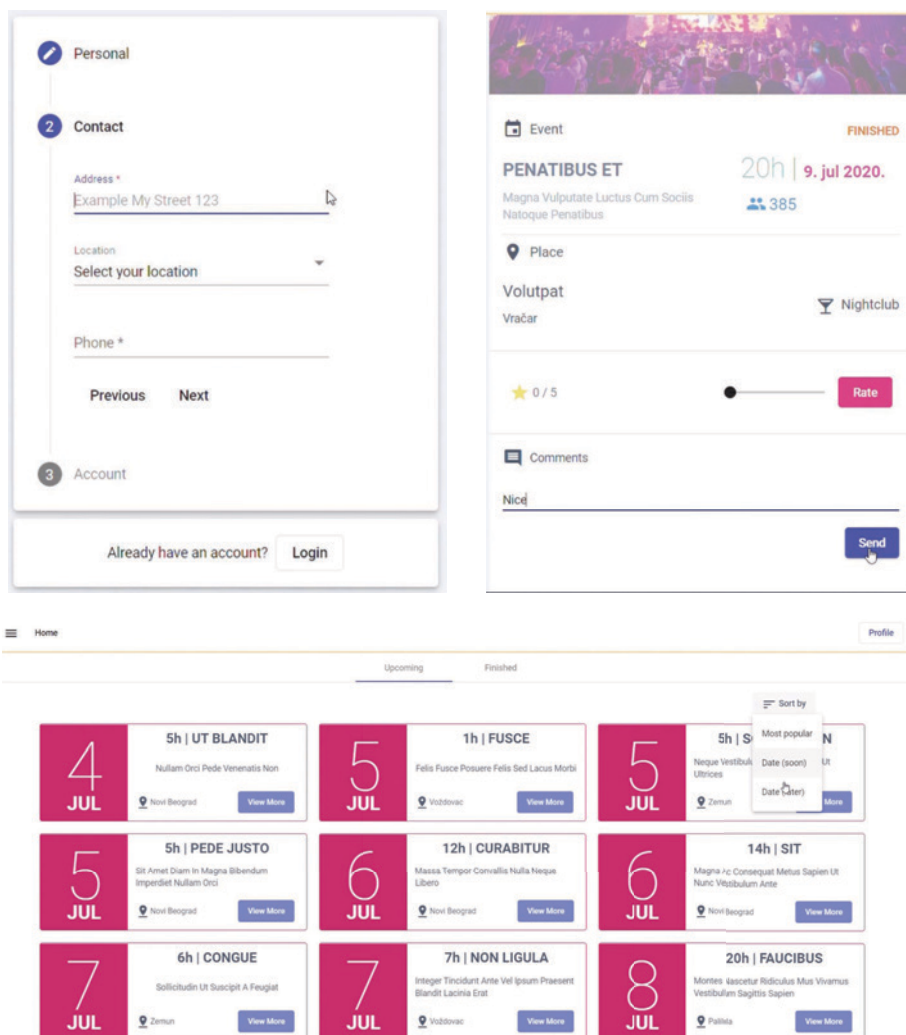


Slika 77. Horizontalni prototip korisničkog interfejsa visoke verodostojnost kreiran korišćenjem alata za prototipiziranje *Mockplus* [34]. Prototip dostupan na: <https://run.mockplus.com/SowmvpE1Ju701AjK/index.html> (15.11.2020).

Sa jedne strane, ovi alati mogu kreirati interaktivne verzije interfejsa koje u velikoj meri podsećaju na krajnji rezultat. Sa druge strane, u obzir se moraju uzeti i njihova ograničenja u pogledu nepostojanja ili nemogućnosti simulacije poslovne (pozadinske) logike aplikacije. Ova ograničenja mogu uticati na doživljaj interaktivnosti interfejsa kod korisnika.

Testiranje računarskog prototipa visoke verodostojnosti nam, uz informacije koje možemo dobiti iz papirnog prototipa, može otkriti sledeće detalje:

- Raspored sadržaja prikaza,
 - Da li su elementi interfejsa dobro pozicionirani i grupisani.
- Detalji sadržaja prikaza,
 - Boje, oblici, veličine, fontovi, ikone (simboli).
- Interaktivna logika interfejsa,
 - Navigacija između stranica, otvaranje/zatvaranje elemenata interfejsa, i povratna informacija za korisnika.
- Detalji upotrebljivosti interfejsa,
 - Lakoća učenja, efikasnost i greške.



Slika 78. Horizontalni prototip korisničkog interfejsa visoke verodostojnost kreiran korišćenjem Angular okvira za razvoj Veb korisničkih interfejsa [37].

7.4. Ciljevi i analiza prototipa

Primarna svrha prototipa jeste brzo kreiranje reprezentacije finalnog proizvoda tako da se može testirati. Drugim rečima, realizacija ideje koja je i vidljiva i merljiva. U tom kontekstu prototipiziranje podrazumeva kompromis između resursa uložених u kreiranje prototipa, i rezultata koji omogućava testiranje sa krajnjim korisnicima. Da bi kompromisi bili lakše i preciznije realizovani, definisani su konkretni ciljevi prototipa kao atributi budućeg proizvoda koji se

žele ispitati, odnosno testirati (Tabela 3). Na ovaj način se složen kompromis vidljivosti i merljivosti posmatra kroz konkretne elemente interfejsa na koje se dizajneri mogu koncentrisati pri izradi prototipa.

Tabela 3. Atributi prototipa formulisani kao specifični ciljevi prototipa.
Preuzeto i adaptirano iz [52].

Atribut	Primeri
Izgled (eng. <i>appearance</i>)	veličina, boja, oblik, margine, oblici
Podaci (eng. <i>data</i>)	veličina podataka, tipovi podataka (brojevi, tekst, grafika), upotreba podataka, organizacija podataka
Funkcionalnost (eng. <i>functionality</i>)	funkcije potrebne korisniku
Interaktivnost (eng. <i>interactivity</i>)	ulazne akcije korisnika, izlazne akcije prototipa, povratna informacija prototipa
Prostorna struktura (eng. <i>spatial structure</i>)	raspored elemenata interfejsa, relacije između elemenata interfejsa

Dva opšta kompromisa u kreiranju jesu verodostojnost širine i dubine. Oni određuju i ciljeve prototipiziranja kao horizontalno i vertikalno prototipiziranje (Tabela 4). Horizontalno prototipiziranje kreira širi skup funkcija niskog nivoa detalja. Ovde se može govoriti o prototipima korisničkog interfejsa. Vertikalno prototipiziranje kreira užu skup funkcija koje su detaljnije realizovane. Ovde se misli na prototipe aplikacija koje realizuju deo njenih funkcija i deo interfejsa za njihovo korišćenje.

Tabela 4. Uporedna analiza prototipa niske i visoke verodostojnosti. Preuzeto i adaptirano iz [52].

Tip	Opis	Prednosti	Mane
Niska verodostojnost	Gruba skica proizvoda. Šematski, približan prikaz sa fokusom na koncepte, ne detalje. Bez ili sa vrlo ograničenom interaktivnošću.	Niža cena izrade. Testiranje alternativa dizajna (veći broj prototipa jedne ideje). Komunikacija korisnika i dizajnera. Otkrivanje problema izgleda i rasporeda interfejsa. Identifikacija novih zahteva korisnika ili tržišta.	Ograničen nivo detalja sadržaja i ponašanja. Ograničena interaktivnost. Ograničena navigacija. Nemogućnost otkrivanja grešaka u korišćenju. Nemogućnost detaljnijeg testiranja upotrebljivosti.
Visoka verodostojnot	Simulacija finalnog proizvoda. Detalji grafičkog dizajna (oblik, boja, veličina, raspored). Interaktivnost sa simulacijom pozadinske logike (bez potpune implementacije poslovne logike).	Kompletnija funkcionalnost. Visoka interaktivnost. Dobra navigacija. Detaljnije testiranje upotrebljivosti. Sličnost sa krajnjim proizvodom (pogodni za svrhe promocije i prodaje).	Skuplji za kreiranje u pogledu resursa u vremena. Nisu pogodni za testiranje koncepta dizajna. Nisu efikasni u prikupljanju korisničkih zahteva.

8. Testiranje sa krajnjim korisnicima

Testiranje sa krajnjim korisnicima je sastavni deo procesa razvoja interaktivnog sistema. Testiranjem se prikupljaju informacije u vezi sa iskustvima korisnika u korišćenju prototipa, aplikacije, uređaja ili komponente sistema. Opšti cilj testiranja jeste unapređenje interaktivnog sistema koji se razvija. Specifični ciljevi mogu biti upotrebljivost sistema (na primer, u kojoj meri je sistem lak za učenje), ili doživljaji koje korišćenje sistema izaziva kod korisnika poput zadovoljstva, postignuća ili motivacije. U nastavku će testiranje sa korisnicima biti opisivano pretežno u kontekstu upotrebljivosti korisničkog interfejsa sistema.

Rasprostranjena upotreba mobilnih uređaja, poput pametnih telefona, tableta i prenosivih računara, je dovela do migracije i prilagođavanja korisničkih interfejsa ovim platformama. Posledica je da upotrebljivost postaje kritičan faktor u korišćenju aplikacija na ovim uređajima. Problem se može javiti i kao rezultat pretpostavke dizajnera da su krajnji korisnici slični njima i da će im se interfejs svideti. Testiranje služi i kao provera ove percepcije, odnosno da li je njihov dizajn prihvatljiv i odgovara potrebama korisnika.

Postoji veći broj metoda testiranja. Na izbor metode pre svega utiče cilj testiranja. Testiranje se može obavljati na različitim mestima, kao što su laboratorije ili namenski uređene prostorije, domovi korisnika, otvoreni prostori, i radna okruženja. Obično uključuje posmatranje korisnika i merenje njihovih aktivnosti. Postoje i metode koje na zahtevaju krajnje korisnike. One aproksimiraju ili modeluju interakciju korisnika sa interfejsom, kao jeftin i efikasan način za procenu interfejsa sa aspekta korišćenja. Nivo kontrole u toku testiranja je takođe promenljiv - bez kontrole u realnim studijama (eng. *field studies*), ili sa kontrolom koja se odnosi na zadatke koji se obavljaju u okruženju u kontrolisanim studijama (eng. *control studies*).

U nastavku poglavlja će biti opisani motivi, predmeti, okruženja i vrste testiranja sa krajnjim korisnicima.

8.1. Principi i vrste testiranja

Pravilno testiranje sa korisnicima, osim razumevanja značaja, podrazumeva i sagledavanje elemenata koji se testiraju, okruženja u kojem se testira, i vremena (trenutka) testiranja u toku razvojnog procesa.

Sa unapređenim mogućnostima tehnologije, rastu i očekivanja krajnjih korisnika. Korisnici, pored toga što očekuju upotrebljiv sistem, žele i da se prijatno osećaju [53]. Samim tim je i značaj testiranja sa korisnicima sve veći.

Sa aspekta reklamiranja i prodaje interaktivnog proizvoda, dobro dizajnirani proizvodi imaju povećanu verovatnoću uspeha. Ovo može opravdati ulaganje kompanije u testiranje sa krajnjim korisnicima. Na ovaj način dizajneri mogu otkriti prave problema i fokusirati se na potrebe ciljne grupe korisnika, umesto da međusobno analiziraju i diskutuju o proizvodu na osnovu sopstvenih očekivanja.

Mogu se testirati prototipi niske verodostojnosti, finalne verzije sistema pre početka operativne upotrebe, specifične funkcije sistema ili zadaci korisnika, estetski aspekti korisničkog interfejsa ili elementi vezani za sigurnost korišćenja.

Predmeti (ciljevi) testiranja su raznovrsni. Na primer, dizajneri aplikacije za pretragu žele da testiraju da li korisnici brže i kvalitetnije pronalaze informacije od interesa. Sa druge strane, dizajneri aplikacije koja promoviše fizičku aktivnost žele da ispituju promene u ponašanju i navikama korisnika. Dizajnere video igre prvenstveno zanima u kojoj meri je igra zabavna korisnicima i koliko dugo će je koristiti. Ustanove za održavanje javnog reda i mira će ispitati da li nova aplikacija koja građanima omogućava prijavu protivzakonitih aktivnosti povećava broj rešenih slučajeva i redukuje ove aktivnosti. Slično tome, kompanija za građevinske radove istražuje da li novi dizajn Veb sajta utiče na povećane broja poslova.

Okruženje u kojem se obavlja testiranje zavisi od predmeta testiranja. Određeni elementi korisničkog interfejsa, kao što su navigacija i orijentacija se testiraju u kontrolisanim ili laboratorijskim uslovima. Razlog je akcenat na specifičnim elementima kako bi se regulisali faktori koji mogu uticati na korišćenje tih elemenata. Upotrebljivost se mahom testira u kontrolisanim uslovima.

Sa druge strane, elementi doživljaja korisnika se uglavnom testiraju u prirodnom, odnosno spontanom okruženju korisnika bez kontrole. Na primer, korišćenje video igara u smislu vremena i emocija koje izaziva. Popularan naziv za ovaj tip testiranja je "studije u divljini" (eng. *in the wild studies*). Na primer, udaljena analiza ponašanja korisnika u aplikacijama društvenih mreža se može vršiti u prirodnom okruženju, gde veća grupa korisnika može boraviti u svojim domovima. Specijalan slučaj prirodnog okruženja su takozvane stvarne laboratorije (eng. *living labs*) kao namenski prostori. Ovi prostori su dizajnirani kao prirodna, svakodnevna okruženja korisnika (poput mesta stanovanja). Sa druge strane, oni poseduju diskretno integrisane senzore koji prate i snimaju aktivnosti i ponašanje korisnika. Koriste se u dizajnu sistema specifičnih namena, poput igara za fizičko vežbanje kod starijih osoba (eng. *exergames*) [54]. U tim slučajevima je od presudnog značaja potpun i precizan uvid u aktivnosti korisnika i potrebe koje budući sistem treba da podrži. U dizajnu ovih

prostora je kritičan balans između prirodnog i eksperimentalnog kako bi korisnici imali doživljaj prirodnog okruženja, a ne da su posmatrani.

Trenutak, odnosno vreme testiranja u toku životnog ciklusa razvoja interaktivnog sistema ponajviše zavisi od vrste sistema. Na primer, to može biti novi sistem ili unapređena verzija postojećeg sistema. U slučaju novog sistema, obično se najpre vreme investira u definisanje korisničkih zahteva (uključujući i istraživanje tržišta). Nakon toga se kreiraju skice proizvoda koje se testiraju rano kako bi se stekao uvid u to da li su dizajneri pravilno razumeli i realizovali zahteve korisnika. U zavisnosti od povratne informacije testiranja, rani prototipi će biti modifikovani i detaljnije razrađeni za naredna testiranja.

Testiranja koja se vrše u toku procesa dizajna kako bi se osiguralo da proizvod nastavlja da zadovoljava korisničke zahteve su poznata kao formativna evaluacija (eng. *formative evaluation*). Ovo testiranje se obavlja u značajnom delu procesa dizajna, od prvih prototipa do završnih izmena konačnog proizvoda.

Testiranja koja se vrše kako bi se procenio završeni proizvod (pre distribucije krajnjim korisnicima) su poznata pod nazivom sumativna evaluacija (eng. *summative evaluation*). Ukoliko se radi o modifikaciji postojećeg sistema tada akcenat nije na uspostavljanju korisničkih zahteva, već na postojeće funkcije sistema kako bi se ustanovilo koje elemente je potrebno unaprediti. Mogu se dodati novi elementi što može proizvesti nove probleme upotrebljivosti, ili se mogu poboljšati postojeći elementi (na primer, raspored elemenata, nazivi kontrola interfejsa ili navigacija).

Ako govorimo o vrstama testiranja sa korisnicima, na osnovu uslova u kojima se odvijaju, angažovanja korisnika i stepena kontrole, možemo razlikovati tri osnovna tipa:

- Kontrolisani uslovi sa korisnicima (poput laboratorija ili stvarnih laboratorija) - aktivnosti korisnika su kontrolisane u cilju testiranja postavljenih hipoteza, odnosno posmatranja i merenja specifičnih ponašanja. Glavne metode su testiranje upotrebljivosti (eng. *usability testing*) i eksperimenti (eng. *experiments*).
- Prirodno okruženje sa korisnicima (poput onlajn zajednica (eng. *online communities*) i sistema koji se koriste u javnim prostorima) - aktivnosti korisnika nisu kontrolisane ili su kontrolisane u manjoj meri kako bi se stekao uvid o korišćenju sistema u realnim situacijama. Osnovni metod su realne studije.
- Okruženje bez korisnika - dizajneri i eksperti kritički analiziraju, predviđaju i modeluju aspekte korisničkog interfejsa kako bi identifikovali probleme upotrebljivosti. Glavne metode obuhvataju heurističku evaluaciju (eng. *heuristic evaluation*), analitiku podataka

(eng. *data analytics*), kognitivni obilazak (eng. *cognitive walkthrough*) i modele (eng. *models*).

Svaki tip donosi određene prednosti i mane. Na primer, testiranja u laboratoriji su pogodna za otkrivanje problema upotrebljivosti, ali ne pružaju uvid u širi kontekst korišćenja i specifične faktore koji mogu uticati na korišćenje. Drugim rečima, kontrolisane studije omogućavaju testiranje hipoteza u vezi sa elementima interfejsa tako da se konkretni rezultati mogu generalizovati i odnositi na širu populaciju.

Realne studije daju dobru predstavu o tome kako korisnici koriste tehnologiju u prirodnom okruženju, ali su skupe i složene za realizaciju. Mogu se prikupiti podaci koji daju uvid u percepcije i doživljaje korisnika u vezi sa korišćenjem novih tehnologija u njihovom svakodnevnom okruženju. Heuristička evaluacija je efikasna i jeftina za implementaciju, međutim može prevideti probleme upotrebljivosti i osetljive aspekte doživljaja krajnjih korisnika.

Osnovni kriterijum za izbor tipa jeste stepen kontrole korišćenja interfejsa od strane korisnika. Preciznije, izbor specifičnih aktivnosti ili zadataka koje će korisnik obavljati na korisničkom interfejsu kako bi stekli uvid o načinu korišćenja sistema.

U nastavku je detaljnije opisan svaki tip testiranja ponaosob.

8.1.1. Kontrolisani uslovi sa korisnicima

Ovaj tip omogućava testerima da kontrolišu šta korisnici rade (koje zadatke obavljaju), kada to rade i u kom trajanju (koliko dugo). Na taj način eliminišu ili redukuju spoljašnje uticaje i ometanja, poput notifikacija ili razgovora sa drugima. Predstavlja široko korišćen i ustaljen način testiranja softverskih aplikacija na stonim ili prenosivim računarima gde korisnici izvršavaju skup zadataka na interfejsu koje je prethodno definisao tester (ili ispitivač).

Testiranje upotrebljivosti je najrasprostranjeniji i opšteprihvaćeni metod testiranja korisničkog interfejsa. Podrazumeva prikupljanje podataka korišćenjem ili kombinovanjem različitih tehnika (posmatranje, upitnik, intervju, logovanje podataka) u kontrolisanom okruženju. Obavlja se u unapred pripremljenoj prostoriji koja ne mora biti laboratorija u smislu specifične opreme i infrastrukture, već može posedovati računar, mesto za opremu i sedenje (sto i stolice), i opremu za snimanje (ukoliko se koristi namenska oprema, ne integrisana u uređaj na kojem se aplikacija izvršava poput kamere, mikrofona ili programa za snimanje ekrana). Po potrebi se može obavljati i sa udaljenih lokacija, sa korisnicima koji su onlajn u prirodnom okruženju (dom ili

posao). Glavni cilj jeste procena upotrebljivosti interfejsa za ciljnu grupu korisnika tako što grupa izvršava zadatke za koje je interfejs namenjen. Ovo podrazumeva ispitivanje načina na koji tipični korisnici obavljaju tipične zadatke na korisničkom interfejsu. Pod tipičnim korisnicima podrazumevamo cilju grupu korisnika, onu kojoj je sistem namenjen. Tipični zadaci su oni koji će korisnici obavljati korišćenjem interfejsa njima namenjenog sistema. Često obuhvata poređenje broja i vrsta grešaka koje se javljaju tokom korišćenja interfejsa, i merenje vremena izvršenja specifičnih zadataka na interfejsu. U toku izvršavanja zadataka, korisnici se mogu snimati. Na primer, direktno korišćenjem kamere, snimanjem ekrana korisnika, ili logovanjem podataka u pozadini. Razlog je prikupljanje podataka radi kasnije analize. Koriste se i upitnici i intervjui kako bi se ispitali faktori koji utiču na zadovoljstvo i doživljaj korisnika u vezi sa korišćenjem sistema.

Posmatranje i analiza reakcija korisnika na korišćenje interaktivnog sistema pomaže u razumevanju i rešavanju problema upotrebljivosti čije je bi otkrivanje bilo složeno analizom izveštaja ili članaka. Istorijski posmatrano, testiranje upotrebljivosti je najrasprostranjeniji metod testiranja i ima poseban značaj za oblast IČR. Unapredilo je razvoj i napredak mnogih standardnih tehnologija kao što su interfejsi mobilnih aplikacija, programi za obradu teksta, alati za matematičke operacije i drugi [29, 50]. Rezultati testiranja upotrebljivosti se često koriste za kreiranje specifikacije upotrebljivosti (eng. *usability specification*) kao zahteva upotrebljivosti za specifičnu vrstu sistema. One se koriste kao smernice za testiranje upotrebljivosti sistema date vrste.

Testiranje se obično vrši u istraživačkim laboratorijama univerziteta, instituta ili kompanija u kojima se ispituju postavljene hipoteze kao specifične pretpostavke korišćenja koje je potrebno verifikovati. Uslovi testiranja se kontrolisani kako bi tester eliminisali faktore koji nisu relevantni za izvršenje zadataka na interfejsu a mogu uticati korišćenje interfejsa. Na taj način se povećava pouzdanost (eng. *reliability*) rezultata u smislu da su oni u najvećoj meri određeni elementima interfejsa a ne drugim faktorima. Na primer, testiranje može vršiti poređenje različitih načina unosa teksta na dodirnom interfejsu mobilnog telefona kako bi se identifikovao najoptimalniji način. Kontrolisani uslovi obuhvataju identične instrukcije za svakog ispitanika, korišćenje identičnog korisničkog interfejsa, i isti zadaci za sve učesnike testiranja. Uslovi koji se porede mogu biti diskretan unos teksta kucanjem tastera dodirne tastature ili kontinualan unos prevlačenjem preko tastera tastaure kako bi se selektovala slova. Cilj jeste utvrditi koji način (uslov) bolji u pogledu brzine unosa i broja grešaka. Određen broj učesnika bi pristupio testiranju odvojeno, izvršavajući skup zadatak unosa teksta u toku kojih bi se merilo vreme obavljanja zadataka i greške koje bi se pojavile (na primer, unos pogrešnog slova ili znaka). Prikupljeni podaci bi se analizirali kako bi se

uporedili rezultati za dva uslova (načina unosa teksta). Ukoliko bi rezultati pokazali da kontinualan unos zahteva kraće vreme i uzrokuje manj broj grešaka, moglo bi se tvrditi da je ovaj način unosa bolji i prikladniji u odnosu na diskretan unos.

8.1.2. Prirodno okruženje sa korisnicima

Glavni cilj realnih studija je testiranje korišćenja u prirodnim, spontanim okolnostima korisnika. Specifični ciljevi mogu biti definisanje zahteva za novim sistemom, uvođenje novog sistema u upotrebu, ili korišćenje postojećeg sistema u novim situacijama. Tipične tehnike uključuju posmatranje, intervjue, fokus grupe i logovanje podataka. Podaci se prikupljaju u obliku događaja i razgovora koji se čuvaju kao ručne beleške (eng. *notes*) ili digitalni audio ili video zapisi. Podatke mogu kreirati i ispitanici u obliku beleški ili dnevnika (eng. *diary*). Sam proces prikupljanja mora biti diskretan i nenametljiv tako da korisnici mogu nesmetano obavljati svoje aktivnosti. Ovo je ujedno i kritičan aspekt zbog toga što na ponašanje korisnika može uticati znanje o tome da su posmatrani. Na ovaj način oni mogu ispoljavati ponašanje koje nije spontano i time uticati na rezultate testiranja [55].

Mobilni uređaji i aplikacije po svojoj prirodi mogu zahtevati ovaj tip testiranja. Na primer, da bi se stekao pouzdan uvid u korišćenje sistema u specifičnim scenarijima u domovima, javnim prostorima i otvorenim prostorima. U takvim okolnostima je smanjen stepen kontrole testiranja od strane istraživača zbog toga što je uticaj kontekstualnih faktora značajan za i korišćenje. Na primer, ukoliko se ispituje funkcija navigacije mobilne aplikacije u urbanom okruženju to se mora obaviti u stvarnim uslovima. Ispitanici bi koristili aplikaciju određeni vremenski period. Istraživač bi najpre uputio ispitanike kako da koriste aplikaciju i nakon toga bi je koristili kada i kako oni žele. Na kraju studije istraživač bi intervjuisao korisnike u vezi sa njihovim iskustvom korišćenja aplikacije.

Ovo istovremeno predstavlja i kritičan aspekt - istraživač nije prisutan čitavo vreme testiranja i može propustiti značajne događaje. U principu, istraživač se oslanja na izveštavanje ispitanika (u obliku ličnih beleški, intervjua i onlajn upitnika). Sa druge strane, ispitanici mogu izostaviti važne detalje, ili koristiti samo određene funkcije iz skupa raspoloživih funkcija. Na taj način istraživač nema pravi uvid u to da li su sve funkcije korisne i upotrebljive. Naravno, činjenica da ispitanici ne koriste određene funkcije može značiti da one nisu korisne, ali to ne mora nužno biti slučaj.

8.1.3. Okruženje bez korisnika

Ovaj tip modeluje ili predviđa ponašanje korisnika prilikom korišćenja sistema koji se testira. Zasniva se na znanju o upotrebljivosti, ponašanju korisnika, situacijama u kojima će sistem biti korišćen, i aktivnostima korisnika. Najpoznatije metode su heuristička evaluacija koja koristi generalizovane smernice o principima upotrebljivosti, i kognitivni obilazak koji definiše jedan ili više tipičnih scenarija kao skupa koraka u korišćenju prototipa sistema.

Heuristička evaluacija koristi heuristike upotrebljivosti koje su originalno razvijene za GKI stonih računara [56]. Vremenom su prilagođavane za potrebe testiranja novih korisničkih interfejsa poput Veb sistema, mobilnih aplikacija, kolaborativnih platformi, tehnologija za zabavu i drugih. Više informacija se može naći u [57].

Kognitivni obilazak simulira proces rešavanja problema (eng. *problem solving*) u koracima dijaloga korisnika i interfejsa. Koraci se analiziraju kako bi se identifikovali problemi u korišćenju interfejsa sistema. Ova metoda prvenstveno otkriva u kojoj meri je interfejs lak za učenje.

Analitika podataka podrazumeva korišćenje logovanja podataka kao tehnike prikupljanja podataka. Često se koristi Veb analitika kao prikupljanje, merenje, analiza i interpretacija podataka u vezi sa onlajn aktivnostima korisnika u cilju razumevanja i optimizacije korišćenja Veb korisničkih interfejsa [45]. Vrsta podataka koja se prikuplja zavisi od ciljeva testiranja. Analitika podataka je korisna za testiranje elemenata Veb sajtova. Mogu se analizirati komentari koje korisnici ostavljaju na sajtu, stranice koje posećuju i vreme koje provode na stranicama, i funkcije koje koriste na korisničkom interfejsu. Koristeći metode automatske obrade podataka, kao što su statistička analiza i vizuelizacija podataka, mogu se otkriti varijacije u Veb saobraćaju, i popularne stranice i servisi. Poseban značaj ima u Veb edukaciji i učenju, odnosno u masivnim otvorenim onlajn kursovima (eng. *Massive Open Online Courses - MOOCs*). U ovim sistemima metoda se koristi kao analitike učenja (eng. *learning analytics*) za procenu efikasnosti i efektivnosti onlajn nastave i učenja.

Modeli se primarno koriste za poređenje efikasnosti različitih interfejsa za obavljanje istih zadataka. Na primer za optimalan raspored i poziciju elemenata korisničkog interfejsa. Poznat je Fitsov zakon (eng. *Fitt's Law*) za procenu vremena pronalaska ciljnog elementa korišćenjem pokazivača [58].

8.2. Relevantni aspekti testiranja

Nezavisno od vrste testiranja, potrebno je obratiti pažnju na dva važna elementa. Pre početka testiranja je neophodno informisati korisnike o testiranju i imati njihov pristanak za učešće u testiranju. Nakon testiranja, treba razmotriti specifične faktore koji mogu uticati na interpretaciju krajnjih rezultata. U nastavku su elementi detaljnije opisani.

8.2.1. Informisanje i pristanak korisnika

Testiranje sa korisnicima spada u grupu aktivnostima koja se obavljaju sa ljudskim bićima. Mnoge organizacije, poput univerziteta, profesionalnih društava, vladinih organizacija i istraživačkih ustanova, zahtevaju informisanje i poštovanje specifičnih procedura u vezi sa takvim aktivnostima. Motiv ovih postupaka jeste zaštita učesnika sa fizičkog aspekta (na primer, kod ispitivanja u medicini), emocionalnog aspekta, kao i aspekta privatnosti u pogledu pravilnog odnosa prema ličnim podacima. U tom smislu, etički odbor institucije koja vrši testiranje (eng. *ethical committee*) na uvid dobija dokumentaciju koja opisuje testiranje i eksplicitno odobrava sprovođenje testiranja. U pitanju je institucija koja angažuje ispitivače. Svaka institucija definiše format i sadržaj dokumentacije u pogledu opisa aktivnosti, podataka koji će se prikupljati, informacije o ispitanicima i slično. Važan deo dokumentacije jeste način na koji će prikupljeni podaci biti analizirani, korišćeni i čuvani. Ovde postoje opšta pravila koja čuvaju privatnost ispitanika. Na primer, podaci koji mogu otkriti identitet ispitanika se moraju kodirati (predstaviti kao oznake ili pseudonimi) i čuvati na bezbednoj lokaciji. U slučaju publikovanja rezultata testiranja, ovi podaci se mogu koristiti kao identifikatori ispitanika.

Nakon dobijanja formalnog odobrenja za sprovođenje testiranja, potrebno je informisati ispitanike i dobiti eksplicitan pristanak pre početka aktivnosti. Konkretno, ispitanike treba upoznati sa procedurom i aktivnostima testiranja, koji podaci u vezi sa njima će biti prikupljeni i kako će biti korišćeni, i njihova prava koja pre svega podrazumevaju povlačenje iz testiranja u svakom trenutku bez obaveze navođenja razloga za takvu odluku. Prethodne informacije se obično pripreme u obliku pisanog ili digitalnog obrasca koji korisnici pročitaju i potpišu pre početka testiranja (u slučaju digitalnog obrasca može se koristiti i element korisničkog interfejsa koji nudi opciju za saglasnost). Ovaj dokument je poznat pod nazivom obrazac pristanka.

8.2.2. Faktori interpretacije rezultata

Vrsta i metoda testiranja u najvećoj meri utiču na vrstu podataka koja će biti prikupljena u toku testiranja. Prilikom analize podataka i interpretacije

rezultata, odnosno definisanja novih saznanja i donošenja zaključaka, moraju se razmotriti i dodatne okolnosti, odnosno faktori koji mogu imati uticaj.

Pouzdanost (eng. *reliability*) ili konzistentnost metode testiranja govori u kojoj meri će metoda dati identične rezultate u drugim, odvojenim prilikama pod istovetnim uslovima. Ispitivač ili istraživač koji prati identičnu proceduru mora dobiti slične rezultate. Različite metode imaju različite stepene pouzdanosti. Na primer, testiranje u kontrolisanim uslovima generalno ima visoku pouzdanost. Sa druge strane, testiranje u prirodnom okruženju ima promenljivu pouzdanost. Nestruktuirani intervju ima nisku pouzdanost zbog toga što je komplikovano i neizvesno sprovesti istovetnu diskusiju sa svakim ispitanikom.

Validnost (eng. *validity*) nam govori da li metoda meri ono čemu je namenjena, odnosno ono što bi trebalo da izmeri. Zavisi od same metode, i načina na koji se ona primenjuje. Na primer, ako je cilj testiranja saznanje o tome kako korisnici koriste nov sistem kod kuće, tada kontrolisani eksperiment nije prikladan. Metode testiranja u prirodnom okruženju su validne u tom kontekstu. Čak i okviru metode, ako na primer želimo da saznamo prosečno vreme izvršenja specifičnog zadatka na interfejsu, metoda koja samo evidentira broj i vrste grešaka prilikom korišćenja nije validna.

Na validnost rezultata testiranja može uticati i takozvani efeket novine (eng. *novelty effect*). Prilikom korišćenja nove tehnologije performanse upotrebe mogu biti poboljšane. Međutim razlog ne mora biti napredak u ovladavanju ili primeni potrebnih veština, već povećano interesovanje za tehnologiju sa kojom se korisnici prvi put susreću [59].

Ekološka validnost (eng. *ecological validity*) je vrsta validnosti koja razmatra u kojoj meri okruženje u kojem se vrši testiranje može uticati na rezultate. Na primer, kontrolisani uslovi u laboratoriji znači da ono što ispitanici rade i njihovo ponašanje može biti drugačije u odnosu na ono šta rade i kako se ponašaju na poslu, kod kuće, ili u slobodno vreme. Prema tome, kontrolisano testiranje ima nisku ekološku validnost zbog toga što je manje verovatno da rezultati oslikavaju ono što bi se zaista dešavalo u stvarnom, nekontrolisanom okruženju. Sa druge strane, posmatranje, realne studije ili logovanje podataka se odigravaju u spontanim uslovima i imaju viši stepen ekološke validnosti.

Pitanje ekološke validnosti se takođe javlja i kada su ispitanici svesni da su posmatrani ili na neki način i sami testirani kao učesnici ispitivanja. Fenomen je poznat pod nazivom *Hawthorne* efekat [55]. Posledica je niza eksperimenata u fabrici električnih uređaja 20-tih i 30-tih godina 20. veka (fabrika se zvala *Western Electric Company's Hawthorne*). Eksperimenti su ispitivali promene u trajanju radnog vremena. Na kraju je ustanovljeno da su radnici reagovali pozitivnije ako su imali poseban tretman u odnosu na boravak u

eksperimentalnim uslovima. Slični efekti se javljaju u medicinskim ispitivanjima (eng. *medical trials*). Pacijenti koji konzumiraju placebo (eng. *placebo*) dozu lekova (falsifikovana doza koja ne sadrži lek) mogu pokazati napredak kao posledica dodatne pažnje koja im se posvećuje i zbog koje se bolje osećaju.

Predrasude (eng. *bias*) se javljaju kada su rezultati interpretirani u skladu sa specifičnim uverenjima, očekivanjima, iskustvima ili stavovima. Drugim rečima, prikupljeni podaci mogu biti pod uticajem i nositi u sebi ove subjektivne faktore tako da njihova analiza ne daje objektivne zaključke. Na primer, inženjeri upotrebljivosti koji sprovode heurističku evaluaciju mogu, kao posledica znanja i iskustva, više pažnje obraćati na specifične probleme dizajna u odnosu na krajnje korisnike. Ovo se može odraziti na rezultate testiranja. Ispitivači koji posmatraju korisnike u prirodnom okruženju mogu selektivno prikupljati samo one podatke koje smatraju relevantnim, i to može uticati na rezultate testiranja. Slično tome, u toku intervjua ispitivači mogu nesvesno uticati na odgovore ispitanika tonom glasa, izrazima lica, jezikom tela i formulacijom pitanja. Zbog toga je važno pravilno formulisati pitanja i ponašati se neutralno u toku intervjua kako bi se eliminisale predrasude u odgovorima koje utiču na rezultate.

Generalizacija (eng. *generalizability*) testiranja govori u kojoj meri se rezultati testiranja mogu uopštiti, odnosno generalizovati. Na primer, Fitsov zakon koji služi za testiranje specifičnih ulaznih uređaja, poput dizajna tastature i miša, ima precizno definisan, uzak domen. Primena zakona daje rezultate koji se ne mogu uopštavati i podjednako odnositi na druge vrste dizajna testiranih uređaja.

8.3. Testiranje upotrebljivosti

Upotrebljivost korisničkog interfejsa se testira u kontrolisanim uslovima. Testiranje upotrebljivosti u pripremljenom okruženju omogućava ispitivačima da kontrolišu aktivnosti korisnika, i društvene faktore i uslove u okruženju koji mogu uticati na rad i rezultate ispitanika. Opšti cilj je istražiti da li je sistem upotrebljiv za ciljnu grupu korisnika u obavljanju zadataka za koje je namenjen.

8.3.1. Elementi testiranja upotrebljivosti - tehnike, zadaci i korisnici

Glavni element testiranja upotrebljivosti je prikupljanje podataka o performansama ispitanika u vezi sa obavljanjem skupa zadataka na korisničkom interfejsu sistema. Tehnike prikupljanja podataka su identične onima opisanim u Poglavlju 6. U kontekstu testiranja upotrebljivosti, tehnike se često kombinuju. Na osnovu značenja koje nose, prikupljeni podaci se mogu

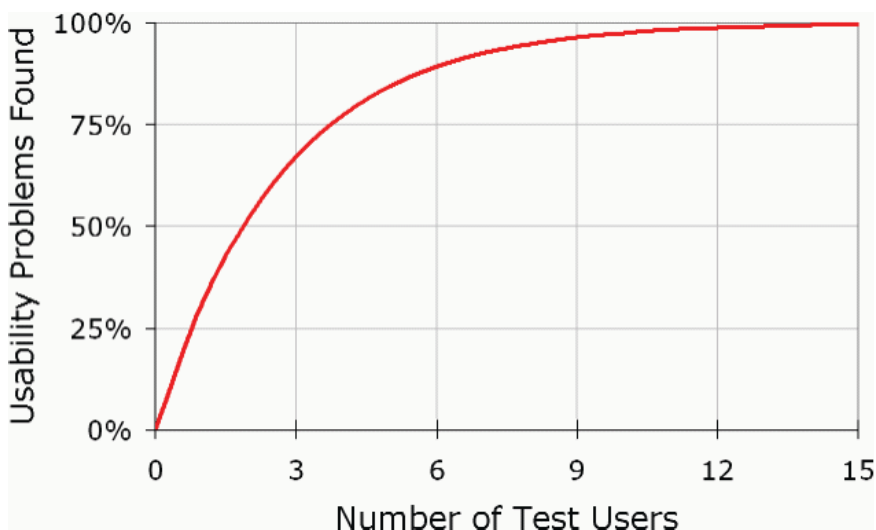
posmatrati kao objektivne i subjektivne mere upotrebljivosti. Objektivne mere su implicitno prikupljeni podaci kao vremena korišćenja, i vrste i količine grešaka koje ispitanici prave. Subjektivne mere su eksplicitno prikupljeni podaci, na primer korišćenjem upitnika i intervjua.

Na primer, ispitanici se mogu snimati radi analize govora i izraza lica. Obično se prate i pokreti miša i akcije na tastaturi. Ispitanici se ponekad i podstiču da glasno razmišljaju (eng. *think aloud*) kako bi se otkrio tok misli i planiranja korisnika dok obavljaju zadatke na interfejsu. Upitnik o zadovoljstvu, koji se obično daje ispitanicima nakon interakcije sa sistemom, otkriva emocionalnu reakciju, odnosno doživljaje koje izaziva korišćenje interfejsa. Varijante intervjua se mogu koristiti za prikupljanje dodatnih informacija u vezi sa tim šta se korisnicima svidelo i šta im se nije svidelo u vezi sa interfejsom.

Primeri zadataka obuhvataju pretragu informacija na korisničkom interfejsu, navigaciju korišćenjem različitih menija, prepoznavanje funkcija na osnovu simbola ikona. Vreme izvršavanja zadatka (na primer nalaženje i izvršavanje akcije na interfejsu) i broj grešaka prilikom izvršavanja zadatka (kao što je selekcija pogrešnog dugmeta ili menija) su najčešće prikupljane mere performansi korisnika. Opšte posmatrano, kvantitativni podaci koji se prikupljaju prilikom testiranja upotrebljivosti mogu dati sledeće informacije:

- Vreme izvršenja zadatka,
- Vreme izvršenja zadatka posle određenog vremena nekorisćenja sistema,
- Broj i tip grešaka po zadatku,
- Broj grešaka u jedinici vremena,
- Broj pristupa ili pokušaja da se nađe pomoć,
- Broj korisnika koji prave specifičnu grešku,
- Broj korisnika koji uspešno izvrše zadatak,
- Subjektivne ocene korisnika (lakoća korišćenja/učenja, zadovoljstvo).

Jedno od glavnih pitanja testiranja upotrebljivosti jeste broj ispitanika koje treba angažovati kako bi rezultati bili pouzdani i verifikovani. Razlog je pre svega praktične prirode, vezan za angažovanje korisnika radi sprovođenja testiranja. Istraživanja pokazuju da je za otkrivanje problema upotrebljivosti dovoljno angažovati između 5 i 12 korisnika (Slika 79) [60, 61]. U praksi se može testirati i sa manjim brojem ispitanika u slučaju ograničenja u budžetu za testiranje ili u planiranju angažovanja korisnika. Na primer, povratna informacija o ideji dizajna na početku procesa razvoja se može dobiti i od 2 ili 3 korisnika kojima je proizvod namenjen. U opštem slučaju, veći broj ispitanika nam daje reprezentativnije rezultate za ciljnu grupu korisnika. Istovremeno će i testiranje zahtevati više resursa i vremena.



Slika 79. Odnos broja ispitanika i identifikovanih problema upotrebljivosti sistema. 5 ispitanika može otkriti 80% ukupnog broja problema upotrebljivosti. Preuzeto iz [61].

Kao oblik ispitivanja u kontrolisanim uslovima, testiranje upotrebljivosti se mora pažljivo i detaljni planirati. Procedura testiranja upotrebljivosti je dokument na osnovu kojeg se sprovodi ispitivanje se korisnicima. Dokument sadrži ciljeve testiranja, opis sistema koji se testira, uslove testiranja (okruženje, materijali, oprema), učesnike (krajnji korisnici), raspored i zadatke za korisnike, i attribute upotrebljivosti koji se ispituju.

Kako bi dobili relevantne i potpune informacije, važno je pravilno definisati zadatke. Svaki zadatak se odnosi na korišćenje jedne specifične funkcije rešenja. Smernice za formulisanje zadataka su sledeće:

- Zadatak mora biti precizan i jasan za korisnika u pogledu terminologije i značenja.
- Zadatak mora biti konkretan i realan,
 - Formulisan kao specifičan cilj korisnika, a ne funkcija sistema koja izvršava zadatak.
- Zadatak ne sme biti formulisan tako da navodi korisnika na rešenje.
- Zadaci se moraju podjednako odnositi na sve relevantne funkcije sistema,
 - Ne samo na one funkcije koje sistem najbolje podržava.
- Zadatak ne sme biti previše fragmentovan,

- Mora ispuniti kompletan cilj koji korisnik postiže njegovim izvršavanjem.
- Pre početka rada na zadacima, naglasiti korisnicima da se testira sistem, a ne oni i da je njihova uloga pomoć u pronalaženju i otklanjanju eventualnih problema.

Slika 80 daje primere zadataka za testiranje upotrebljivosti korisničkog interfejsa karpuling aplikacije čiji su zahtevi opisani u Poglavlju 5.4.

“Zamislite da želite da počnete da koristite aplikaciju. Na koji način biste to uradili?”

(a)

“Pronađite sve vožnje od lokacije Danijelova za 15.12. u vremenu između 13 i 14 sati.”

(b)

“Rezervišite jednu od ponuđenih vožnji.”

(c)

“Ponudite vožnju na relaciji Danijelova-Kumodraška za danas u 15 sati.”

(d)

Slika 80. Primeri zadataka za testiranje upotrebljivosti horizontalnog prototipa karpuling aplikacije za studente čiji su zahtevi opisani u Poglavlju 5.4. Zadaci su definisani sa aspekta ciljeva korisnika i testiraju: (a) funkciju registracije novog korisnika, (b) funkciju pretrage vožnji, (c) funkciju rezervacije vožnje, i (d) funkciju kreiranja vožnje.

Prkupljeni podaci mogu biti kvalitativne i kvantitativne prirode. Vrste podataka su opisane u Poglavlju 6.3. Kvalitativni podaci mogu dati informacije o ponašanju, stavovima i razmišljanju ispitanika (na primer, na osnovu posmatranja i intervjua). Značajni su za otkrivanje razloga ponašanja korisnika i uzroka problema koji se mogu pojaviti u toku korišćenja sistema. Kvantitativni

podaci opisuju aktivnosti korisnika i mogu identifikovati probleme u korišćenju (kao što su vreme izvršavanja zadatka i broj grešaka).

Vrste podataka su komplementarne, odnosno korišćenje tehnika za njihovo prikupljanje nam mogu dati detaljnije i potpunije informacije u vezi sa testiranjem upotrebljivosti. Opšte posmatrano, kvantitativni podaci opisuju pojave vezane za korišćenje interfejsa. Sa druge strane, kvalitativni podaci mogu detaljnije opisati pojave i otkriti razloge njihovog nastanka. U razvoju korisničkog interfejsa početna testiranja su obično kvalitativna sa prototipima niske verodostojnosti. U početku je dovoljno je testirati sa manjim brojem korisnika i modifikovati dizajn na osnovu kvalitativne analize korišćenja sistema. Vreme izvršenja ili broj grešaka u ovoj fazi nisu od presudnog značaja. Kako proces dizajna odmiče protoip postaje detaljniji i počinje se sa kvantitativnim testiranjima nakon što su prethodno otklonjeni problemi korišćenja. U završnim fazama, uključujući i sumativnu evaluaciju, kvalitativne i kvantitativne tehnike se mogu kombinovati (na primer logovanje podataka sa posmatranjem i upitnikom).

Tehnika za prikupljanje kvalitativnih podataka specifična za testiranje upotrebljivosti je glasno razmišljanje. Može otkriti značajne informacije koje nisu neposredno vidljive ili dostupne zbog toga što se podaci vezani za tok misli korisnika u toku obavljanja zadataka (kao što su planiranje, donošenje odluka i zaključivanje) ne mogu prikupiti korišćenjem drugih tehnika. Primena tehnike se svodi na podsticanje korisnika da razmišlja na glas dok izvršava zadatke testiranja i evidentiranje izgovorenih misli (kroz snimanje razgovora ili vođenje beleški). Prikupljeni podaci mogu dati različite informacije - o čemu korisnici razmišljaju dok obavljaju zadatke, šta pokušavaju da urade da interfejsu, problemi na koje nailaze u toku korišćenja, kako vide i doživljavaju korisnički interfejs, i druge. Korisnicima se postavljaju pitanja koja su relevantna za trenutni kontekst korišćenja, odnosno zadatak koji obavljaju na interfejsu. Ispitivač mora biti neutralan - korisniku ne treba pomagati kroz razgovor ili postavljati pitanja koja navode na rešenje, osim u slučaju kada je pomoć zaista potrebna. Razgovor se snima (ili se vode ručne beleške) radi kasnije analize.

Pre testiranja sa ispitanicima, korisno je sprovesti pilot-testiranje procedure i zadataka. Svrha pilot-testiranja jeste verifikacija ili provera definisane procedure testiranja. Konkretno, da li korisnici razumeju zadatke, koliko vremena je potrebno po zadatku, da li su pitanja intervjuja ili upitnika koji se planiraju koristiti razumljiva korisnicima i slično. Ovim se može smanjiti rizik u pogledu grešaka vezanih za prikupljanje podataka u toku testiranja, i ublažiti ili eliminisati uticaj faktora interpretacije rezultata (opisanih u Poglavlju 8.2.2). Na taj način se može obezbediti željeni kvalitet prikupljenih podataka. Pilot-testiranje se može obaviti i sa jednim korisnikom.

U toku testiranja dobro je da atmosfera bude prirodna, ne suviše formalna (kontrolisani uslovi ne podrazumevaju formalnu atmosferu testiranja). Korisnik izvršava zadatak po zadatak, ne veći broj zadataka istovremeno. Zadaci se mogu dati korisniku i u pisanoj formi. Ispitivač mora imati neutralan stav u toku testiranja i ni na koji način ne uticati na korisnika u toku izvršavanja zadataka (verbalno i neverbalno). Testiranje početi sa lakšim zadacima i nastaviti ka složenijim. Korisnik u svakom trenutku može odustati od tekućeg zadatka ili od testiranja u potpunosti bez obrazloženja. Na kraju testiranja se zahvaliti korisniku.

Analiza kvalitativnih podataka se obično sastoji u pronalaženju ponavljajućih pojava u prikupljenim podacima kao obrazaca ili tema [47]. Ove teme predstavljaju kritične pojave (eng. *critical incidents*) koje utiču na korišćenje sistema. Mogu biti pozitivne, ističući dobre aspekte sistema, ili negativne, ističući elemente koje je potrebno unaprediti ili probleme koji se moraju otkloniti. Kod testiranja upotrebljivosti akcenat je na otkrivanju i analizi negativnih kritičnih pojava u cilju poboljšanja sistema koji se razvija. Primeri uključuju izbor pogrešne opcije (dugmeta ili ikone) za pokretanje željene funkcije, specifična akcija se ne može pronaći na interfejsu, redosled koraka zadatka koji uzrokuje greške, ili funkcija koju korisnik ne ume da koristi.

U zavisnosti od količine i uticaja kritičnih pojava na korišćenje aplikacije, one se mogu rangirati na osnovu ozbiljnosti (eng. *severity level*). Obično se definišu tri nivoa ozbiljnosti kao nizak (eng. *low*), srednji (eng. *middle*) i visok (eng. *high*). Kriterijumi za definisanje nivoa mogu biti uticaj kritične pojave na dalje korišćenje sistema i broj ispitanika kod kojih se javlja pojava. Prioritet u rešavanju imaju kritične pojave visoke ozbiljnosti. Rešavanje ozbiljnijih pojava može kao rezultat imati i otklanjanje manje ozbiljnih problema.

Kvantitativna analiza prikupljenih podataka testiranja može biti korisna za dobijanje specifičnih informacija kao što su vremenski zahtevi izvršenja zadataka, procenat uspešnosti izvršenja zadatka, ili poređenje performansi različitih sistema u pogledu brzine i/ili grešaka.

Ukoliko se različite tehnike prikupljanja kombinuju treba imati na umu da glasno razmišljanje ponekada može ometati ispitanika sa aspekta brzine i preciznosti korišćenja interfejsa.

Testiranje upotrebljivosti može dati visoko varijabilne podatke u pogledu performansi korisnika. Razlog su prirodne razlike u osobinama, veštinama i znanjima ispitanika. Samim tim je potrebna je veća količina podataka za validnije i pouzdanije rezultate. Ovo zahteva angažovanje većeg broja ispitanika što dalje iziskuje dodatne troškove u pogledu organizacije i sprovođenja testiranja.

Na primer, ako merimo vreme izvršavanja zadataka na interfejsu, angažujemo N=6 ispitanika i kao rezultat dobijemo vremena od 20, 15, 40, 90, 10, 5 minuta

po korisniku. Računajući srednju i središnju vrednost uzorka (*mean* i *median*, respektivno) dobijamo vrednosti od 30 i 17.5 minuta. Međutim, rezultat kao takav nije reprezentativan zbog toga što za relativno mali broj korisnika postoji visoka standardna devijacija ($SD=32$) vremena izvršavanja zadataka. Visoka varijabilnost govori da uzorak nije reprezentativan i da imamo korisnike sa velikim razlikama u znanjima i veštinama relevantnim za korišćenje sistema. Jedan od ključnih aspekata kvantitativnog testiranja upotrebljivosti jeste odgovarajući balans broja korisnika i rezultata testiranja. Drugim rečima, potrebno je odrediti dovoljan broj ispitanika koji može dati pouzdane rezultate testiranja. Stvarnu varijabilnost korisnika je potrebno svesti na reprezentativni uzorak ispitanika za testiranje čije angažovanje može dati zadovoljavajuće rezultate.

Kvantitavni podaci generalno daju statistički pouzdanije rezultate sa većim brojem korisnika. Praksa kvantitativnog testiranja upotrebljivosti Veb korisničkih interfejsa je pokazala da rezultati prate normalnu raspodelu (eng. *normal distribution*) [62]. Ovo može olakšati i poboljšati analizu rezultata tako što se definišu granični skupovi vrednosti van opsega koji se posmatra (eng. *outliers*). Sa aspekta statističke analize, smatra se da rezultati korisnika koji pripadaju graničnim skupovima nisu reprezentativni. Na primer, to mogu biti korisnici koji suviše sporo ili brzo izvršavaju zadatke na interfejsu. Kao takvi, njihovi rezultati mogu uticati na srednje vrednosti i interpretaciju rezultata, i zbog toga se ne razmatraju u statističkoj analizi. Sa druge strane, rezultate ovih korisnika treba razmotriti i ustanoviti prave razloge njihovih performansi prikupljanjem kvalitativnih podataka (korišćenjem intervjuva ili upitnika).

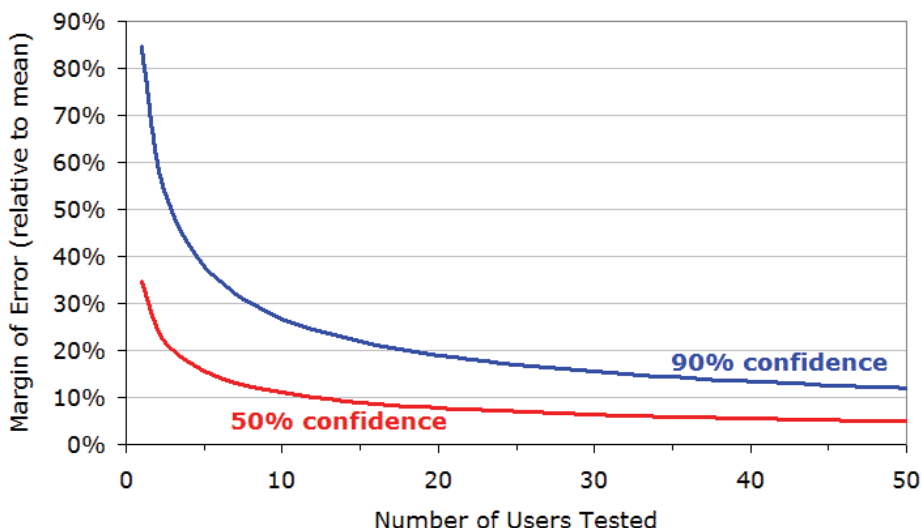
Interval pouzdanosti (eng. *confidence interval*) preciznije određuje deo rezultata koji će biti razmatrani u odnosu na srednju vrednost (*mean*). Definisan je marginom greške (eng. *margin of error*) kao dozvoljeno odstupanje rezultata (izraženo u procentima) od srednje vrednosti. Na taj način, očekivana, odnosno prihvatljiva vrednost se nalazi unutar intervala čiji je središnja vrednost srednja vrednost, dok su donja i gornja granice određena vrednošću margine. *Opšteprihvaćena vrednost praga pouzdanosti za testiranje upotrebljivosti Veb korisničkih interfejsa je 90 procenata* [62]. Može se reći i da će 90% vremena rezultati biti u ovom opsegu (očekivana brzina rada korisnika), 5% će biti u donjem graničnom opsegu (suviše brzi korisnici), i 5% u gornjem graničnom opsegu (suviše spori korisnici).

Prag pouzdanosti i margina greške mogu dati informaciju o dovoljnom broju korisnika za kvantitativno testiranje upotrebljivosti korisničkog interfejsa [61]. Slika 81 daje zavisnosti margine greške od broja ispitanika za specifične vrednosti praga pouzdanosti, crvena kriva za vrednost od 50% i plava kriva za vrednost od 90%, respektivno. Zavisnosti na slici se odnose na prikupljanje različitih mera upotrebljivosti (najčešće su to vreme izvršenja i broj grešaka). Pored toga, određene su i pravilom da će prilikom kvantitativnog testiranja

upotrebljivosti približno 6% korisnika kreirati rezultate u graničnim opsezima (koji neće biti razmatrani) [63].

Za relaksiranu vrednost praga pouzdanosti (crvena boja) vrednosti graničnih intervala su isuviše visoke (po 25%) i samim tim nam je potreban veći broj ispitanika kako bi dobili pouzdanije rezultate zbog toga što se rezultati korisnika u graničnim intervalima ne razmatraju.

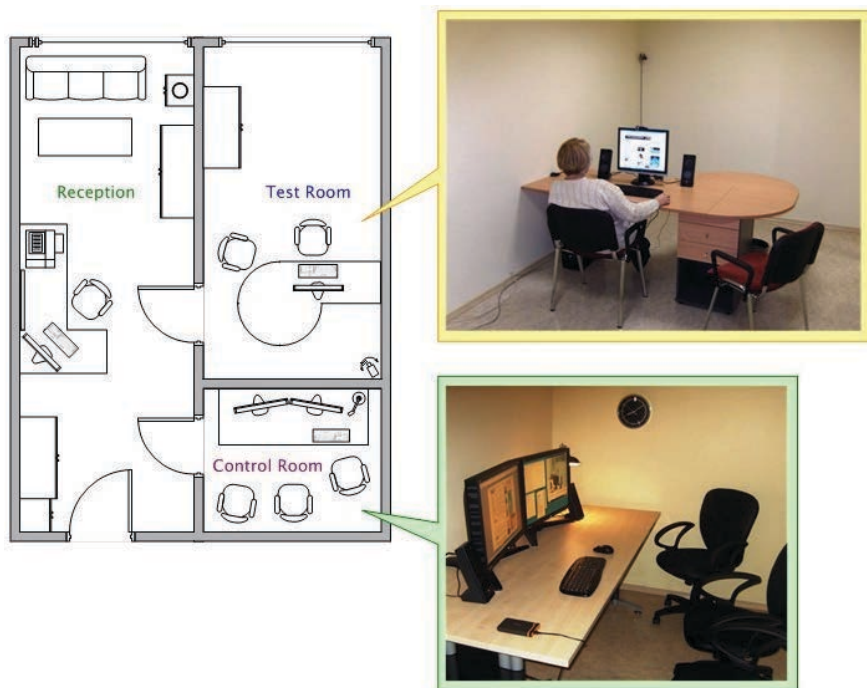
Ako analiziramo plavu krivu (veći prag pouzdanosti) za $N=20$ ispitanika, margina greške iznosi 19% ispod i iznad srednje vrednosti rezultata. Na primer, ukoliko posmatramo vreme izvršenja zadatka i srednja vrednost iznosi 4 minuta, margina greške je ± 46 sekundi i interval pouzdanosti iznosi od 194 sekunde do 286 sekundi. *U praksi kvantitavnog testiranja upotrebljivosti $N=20$ korisnika se smatra dovoljnim za dobijanje pouzdanih rezultata sa aspekta statističke obrade* [62]. Na slici ispod se može primetiti da dalje povećanje broja ispitanika ne utiče značajnije na promene vrednosti margine greške, dok uvećava troškove testiranja.



Slika 81. Kvantitativno testiranje upotrebljivosti - odnos broja ispitanika i margine greške za specifične vrednosti pouzdanosti podataka. 20 korisnika može dati statistički pouzdane rezultate kvantitativnog testiranja upotrebljivosti. Preuzeto iz [61].

Brojevi korisnika potrebni za otkrivanje problema upotrebljivosti ($N=5$, Slika 79) i statistički pouzdano testiranje upotrebljivosti ($N=20$, Slika 81) govore da početna testiranja trebaju biti kvalitativna sa manjim brojem ispitanika, dok se za kasnija kvantitativna testiranja može angažovati veći broj ispitanika.

Za potrebe testiranja se mogu koristiti namenski kreirane laboratorije za testiranje upotrebljivosti (Slika 82a) ili tiha prostorija sa računarom na kojem se koristi interfejs i opremom za snimanje (Slika 82b).



(a)



(b)

Slika 82. Prostorije za testiranje upotrebljivosti - (a) Laboratorija, odnosno eksperimentalno okruženje za testiranje upotrebljivosti. Izvor: <https://interux.com/Usability-Lab-Estonia-Tallinn.html>; (b) Prostorija sa računarom. Izvor: <https://www.testingtime.com/en/blog/nine-golden-rules-solid-user-testing/> (15.11.2020).

Prezentacija rezultata testiranja podrazumeva izveštaj (materijalni ili digitalni). Izveštaj sumira i kritički analizira rezultate ispitanika u vezi sa prethodno definisanim ciljevima, odnosno posmatranim merama upotrebljivosti. Tvrdnje u vezi sa rezultatima moraju biti potkrepljene podacima. Kritični incidenti mogu biti ilustrovani i citatima korisnika. Izveštaj može biti dodatak procedure testiranja upotrebljivosti. Vizuelni elementi poput tabela i dijagrama mogu doprineti bržem i boljem razumevanju rezultata. U nastavku je dat njihov opis sa tipičnim slučajevima korišćenja.

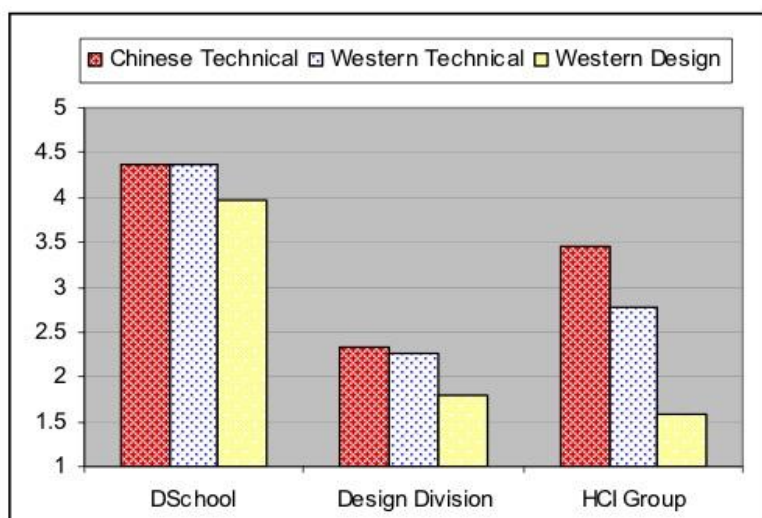
Tabelarni prikazi su pogodni za sumiranje rezultata testiranja, tako što kolone sadrže mere ili kategorije, dok redovi daju vrednosti, odnosno instance mera ili kategorija. Slika 83 daje rezultate testiranja upotrebljivosti za deo korisničkog interfejsa aplikacije. Rezultati prikazuju greške u izvršavanju specifičnih funkcija dela interfejsa. Prilikom testiranja upotrebljivosti tehnike prikupljanja podataka se mogu kombinovati. U primeru u tabeli korisnici su nakon izvršenja zadatka popunjavali kratak upitnik u kom su opisivali problem i predlagali rešenja (prikazani u krajnjoj koloni).

Table 3. Groups critical controls in terms of errors, correct actions and comments.

Severity	Control	Errors	Correct	Comments (number)
High	Membership mechanism	10	0	Add the ownership and member invitation that includes user search function (7).
High	Pen icon (groups tab)	10	0	Recognised to create a group, but creates a story. Should be removed because there is "Create group" button (10).
High	Localisation	0	10	Not fully translated into Italian, for example "Choose file" button (6).
Medium	Feedback messages	0	10	Too short on smartphones and tablet and interferes with the menu bar colour (5).
Low	Buttons	0	10	Use always green background and white text because it is also easier to differentiate (3).
Low	Search for existing groups	3	0	Would like to have this option as part of standard search box, in addition to existing search tags (3).

Slika 83. Tabela prikazuje rezultate testiranja upotrebljivosti funkcija društvenih grupa korisničkog interfejsa. Kolone predstavljaju ozbiljnost kritične pojave, element interfejsa na koji se odnose, uspešna i neuspešna korišćenja elementa (greške), i sugestije za unapređenje (broj označava broj korisnika koji su dali komentar).

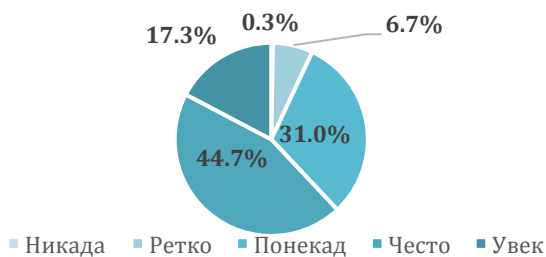
Bar dijagram (eng. *bar diagram*) je pogodan za poređenje vrednosti većeg broja kategorija podataka ili isticanje učestalosti pojava (na primer, preko srednje vrednosti skupa). Slika 84 daje uporedni prikaz srednje vrednosti subjektivnih mera estetskog doživljaja korisničkog interfejsa, prikupljenih upitnikom za različite grupe korisnika nakon testiranja funkcija interfejsa.



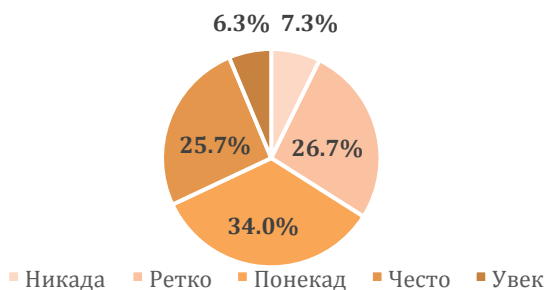
Slika 84. Bar dijagram koji daje uporedni prikaz srednje vrednosti subjektivnog estetskog doživljaja korisničkog interfejsa za različite ciljane grupe korisnika. Preuzeto iz [64].

Pita dijagram (eng. *pie diagram*) je pogodan za prikaz proporcija vrednosti kategorija podataka unutar skupa pojava. Slika 85 ilustruje odnos specifičnih odgovora na pitanja u vezi sa korišćenjem aplikacija društvenih mreža, gde su kategorije vrednosti skale upitnika na slici.

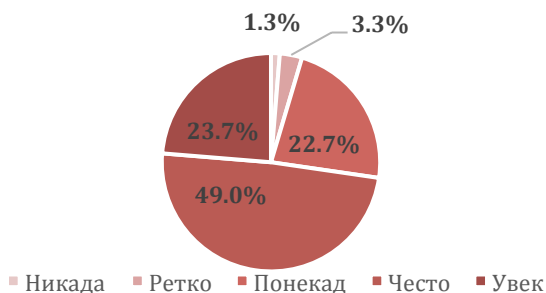
Колико често "одлутате" у коришћењу друштвене мреже након што испуните примарни циљ њеног коришћења?



Колико често заборавите разлог за коришћење друштвене мреже?

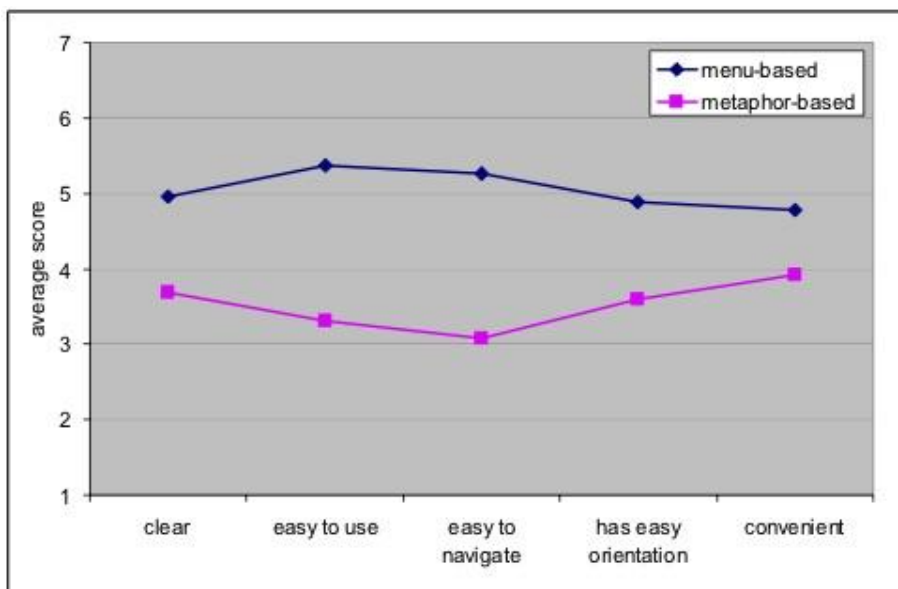


Колико често мислите да беспотребно проводите време на друштвеним мрежама?



Слика 85. Пита дијаграм који приказује однос вредности одговора упитника у вези са коришћењем апликација друштвених мрежа (N=300 испитаника). Преузето из [46].

Linijski dijagram (eng. *plot diagram*) je pogodan za prikaz trenda promene vrednosti specifične nezavisne varijable testiranja. Trendovi se mogu definisati u vremenskom kontekstu ili kontekstu specifičnih zavisnih varijabli testiranja. Slika 86 ilustruje vizuelizaciju promena dve vrednosti nezavisne varijable (GKI koji pretežno koristi menije ili simbole, odnosno ikone kao metafore) posmatranih u odnosu na skup zavisnih varijabli kao srednjih vrednosti specifičnih mera upotrebljivosti.



Slika 86. Linijski dijagram pogoduje praćenju trendova posmatranih vrednosti. Dijagram prikazuje uporedni prikaz kretanja dve vrednosti nezavisne varijable (kao specifični tipovi interfejsa) u odnosu na mere upotrebljivosti kao zavisne varijable. Preuzeto iz [65].

Testiranje upotrebljivosti korisničkog interfejsa podrazumeva tri uloge - korisnik (ispitanik), moderator (eng. *facilitator*) i posmatrač (eng. *observer*). Moderator je ispitivač koji direktno interaguje sa korisnikom. Najpre informiše korisnika o testiranju i dobija eksplicitan pristanak za nastavak studije. Nakon toga korisniku daje zadatke (u pisanoj i/ili usmenoj formi) i prati rad korisnika. U toku rada, po potrebi pomaže korisniku, podstiče ga da glasno razmišlja. Testiranje može kombinovati različite tehnike. Na primer, akcije i podaci korisnika se mogu beležiti i istovremeno se može snimati ekran računara na kojem korisnik izvršava zadatke (eng. *screen capture*). Nakon svakog zadatka korisniku se može dati kraći upitnik u vezi za jednostavnost zadatka i zadovoljstva (na primer, likert skala sa 7 podeoka). Po obavljanju svih

zadataka, korisniku se može ponuditi upitnik otvorenog tipa za komentare i sugestije. Dobra praksa je pitati korisnika da navede pozitivne i negativne elemente u vezi sa obavljanjem zadataka (na primer, po 3 odgovora za element). Na ovaj način se mogu dobiti konkretniji zahtevi za unapređenje upotrebljivosti korisničkog interfejsa. Po završetku rada korisnika, moderator zatvara sesiju.

Posmatrač ima zadatak da prikuplja beleške i prati testiranje. Ova uloga uvek postoji u testiranju upotrebljivosti. U zavisnosti od složenosti i okolnosti testiranja, jedan ispitivač može obavljati jednu ili dve uloge, drugim rečima u isto vreme biti samo moderator ili posmatrač, ili i posmatrač i moderator. U praksi su zastupljena oba pristupa. U slučaju da moderator ne može prikupljati beleške i neprestano pratiti rad korisnika, a da pri tome ne utiče na korisnika, dobro je angažovati još jednog ispitivača kao posmatrača. Posmatrač zapaža i evidentira značajne situacije u radu korisnika kao kritične pojave. Za celokupno vreme testiranja posmatrač mora biti neutralan - ni na koji način ne učestvuje u komunikaciji sa korisnicima i isključivo posmatra njihov rad.

Podaci vezani za rad korisnika se mogu prikupljati na različite načine, Na primer mogu se koristiti i kombinovati vođenje ručnih beleški, audio i video snimanje, i snimanje ekrana.

8.3.2. Elementi metodologije testiranja

Testiranje upotrebljivosti mora biti obavljeno u skladu sa opštim metodološkim principima kako bi rezultati bili verodostojni. Ključni metodološki elementi testiranja upotrebljivosti obuhvataju:

- definisanje subjekata kao reprezentativnog uzorka ciljne grupe korisnika,
- definisanje hipoteza kao relacija između zavisnih i nezavisnih varijabli,
- testiranje hipoteza,
- eksperimentalni dizajn kao specifičan način testiranja hipoteza, i
- elemente statističke analize rezultata.

U nastavku je dat opis elemenata.

8.3.2.1. Definisanje hipoteza

Evaluacija upotrebljivosti podrazumeva testiranje specifičnih hipoteza koje predviđaju performanse korisnika u vezi sa korišćenjem interfejsa. Na primer, hipoteza može biti da kontekstualni meni (čiji je sadržaj određen kontekstom definisanim prethodno izabranom opcijom interfejsa) omogućava lakši izbor opcija od hijerarhijskog (kaskadnog) menija (koji ima fiksni sadržaj). Hipoteze

se obično zasnivaju na teorijskim saznanjima (na primer, Fitsov zakon) ili rezultatima ranijih istraživanja. Specifične metrike služe testiranju hipoteza. U prethodnom primeru, za dve vrste menija se može računati preciznost izbora stavki menija na osnovu broja grešaka koje korisnici prave pri selekciji.

8.3.2.2. Testiranje hipoteza

Hipoteza opisuje relaciju između dve stvari koje zovemo promenljive ili varijable. Varijabla može biti zavisna i nezavisna. Nezavisna varijabla je stvar kojom upravlja, odnosno manipuliše istraživač. U primeru iznad nezavisnu varijablu čine dva tipa menija. Druga stvar se naziva zavisna varijabla i njena se vrednost ispituje u eksperimentu, odnosno testiranju. Na nju utiče vrednost nezavisne varijable. U datom primeru, zavisna varijabla je vreme potrebno za izbor stavke menija. Zavisna varijabla predstavlja meru performanse čija će vrednost varirati u zavisnosti od tipa menija kao vrednosti nezavisne varijante, ukoliko je postavljena hipoteza tačna. Drugim rečima, nezavisnom varijablom upravlja istraživač kako bi verifikovao hipotezu na osnovu vrednosti zavisne varijable.

Prilikom definisanja hipoteze koja ispituje uticaj nezavisne varijable na zavisnu varijablu, moguće je kreirati i nultu hipotezu i alternativnu hipotezu. Nulta hipoteza bi mogla tvrditi da ne postoji razlika u performansama korisnika (vreme izbora stavke) za dva tipa menija. Alternativna hipoteza bi govorila da postoji razlika u performansama korišćenja dva menija. Ukoliko je razlika samo navedena, ali ne i specificirana onda je hipoteza dvostrana zbog toga što može ima dve interpretacije: prvi ili drugi tip menija se može brže koristiti. Slično tome, može se definisati i jednostrana hipoteza, na primer kontekstni meni se brže koristi od hijerarhijskog menija. Izbor jednostrane ili dvostrane hipoteze zavisi od prethodnog uverenja (rezultata ili teorije) koje podržava hipotezu. Razlog korišćenja nulte hipoteze je metodološke prirode - u slučaju razlika u performansama menija nulta hipoteza je opovrgnuta, u suprotnom je potvrđena.

Da bi hipoteza bila testirana, istraživač mora uspostaviti uslove u kojima su druge varijable, koje mogu uticati na rezultate, konstantne u toku testa. Primeri ovih varijabli koje moraju biti identične za oba menija su veličina i rezolucija ekrana. U suprotnom mogu uticati na rezultat. Na primer, ukoliko jedan meni koristi font manje veličine od drugog menija onda ova razlika može uticati na brzinu korišćenja menija, a ne struktura i izgled menija.

Može se testirati i relacija između dve nezavisne varijable, na primer starost i nivo obrazovanja. Hipoteza može tvrditi da mlađe osobe efikasnije (brže) pretražuju korisnički interfejs od starijih osoba i da osobe sa višim nivoom obrazovanja efektivnije (kvalitetnije) pretražuju interfejs. Testiranje bi merilo vreme izvršenja zadatka pretrage i broj pretraga. Analiza podataka mora uzeti u

obzir uticaj nezavisnih varijabli (starost i nivo obrazovanja) i istražiti relacije između njih kako bi se ustanovile moguće međuzavisnosti.

Testiranje hipoteza može biti dalje prošireno razmatranjem dodatnih varijabli. Ovo sa druge strane usložnjava sam eksperiment, odnosno testiranje. Prethodni primer može testirati efekte starosti i nivoa obrazovanja za dve metode pretrage korisničkog interfejsa: jedna koja koristi dugme i polje za pretragu, druga kod koje je pretraga ručna (kretanje kroz interfejs izborom opcija). Sada je cilj testiranje uticaja tri nezavisne varijable (starost, nivo obrazovanja, metod pretrage interfejsa) i ispitivanje njihovih međurelacija. Povećanje broja nezavisnih varijabli usložnjava analizu podataka kako bi se otkrili i odredili uticaji svake varijable na rezultat.

8.3.2.3. Eksperimentalni dizajn

Prilikom sprovođenja testiranja potrebno je odrediti koji ispitanici će biti testirani u vezi sa specifičnim uslovima, odnosno hipotezama i nezavisnim varijablama. Učešće u jednom uslovu može uticati na performanse ispitanika u slučaju učešća u drugom uslovu. Na primer, jedna grupa korisnika je učila strani jezik korišćenjem multimedijalnih lekcija i nakon toga koristila iste ili slične edukativne materijale, ali kroz drugačiju aplikaciju (kao što je konverzijski agent). Ukoliko se druga grupa korisnika prvi put susreće sa datim materijalima korišćenjem agenta, tada je prva grupa u prednosti zbog toga što se prethodno susretala, odnosno dodatno koristila dati sadržaj. Ovo bi uticalo na rezultat jednog testiranja u pogledu predrasuda zbog toga što je deo korisnika prethodno bio izložen identičnom gradivu. Ta grupa korisnika ima prednost zbog toga što je imala dodatno vreme da savlada gradivo čime je povećana verovatnoća ispravnih odgovora. U tom slučaju, jedna grupa korisnika koja nije bila izložena materijalima za učenje treba biti angažovana u eksperimentu. Opisana pojava je poznata pod nazivom efekat treninga (eng. *training effect*).

U vezi sa prethodnim razmatranjima, varijante eksperimentalnog dizajna mogu biti:

- između grupa (eng. *between-group* ili *between-subject*),
- unutar grupe (eng. *within-group* ili *within-subject*) i
- uparivanje (eng. *matched-pairs*).

Varijanta *između grupa* znači da je jedna grupa korisnika angažovana u jednom specifičnom uslovu testiranja. Na taj način, različiti korisnici bivaju izloženi različitim uslovima. Na primer, jedna grupa korisnika može vršiti pretragu korišćenjem polja i dugmeta za pretragu, dok druga grupa pretražuje informacije na interfejsu kretanjem kroz interfejs. Prednost ovakvog dizajna je

nepostojanje efekta treninga uslovljenog iskustvom korisnika u jednom skupu zadataka koje može uticati na performanse izvršavanja narednog skupa. Svaki korisnik samo jednom izvršava zadatke. Mana dizajna je što zahteva veći broj ispitanika kako bi grupe bilo ujednačene, odnosno da bi uticaj individualnih razlika između učesnika, u pogledu znanja i veština, bio minimalan. Slučajno raspoređivanje korisnika i prethodno testiranje korisnika, kako bi se uočili ispitanici koji se značajnije razlikuju od ostalih, mogu ublažiti mane.

Varijanta *unutar grupe* podrazumeva angažovanje svih korisnika u svim uslovima testiranja. Na primer, svaki korisnik pretražuje interfejs korišćenjem polja i dugmeta za pretragu, i kretanjem kroz interfejs. Na taj način je potreban manji broj ispitanika. Glavni motiv za ovu varijantu dizajna je eliminisanje uticaja individualnih razlika na performanse korisnika u različitim uslovima, i uvid u varijacije performansi svakog korisnika u različitim uslovima. Kritičan element ovog tipa je efekat učenja (eng. *learning effect*) zbog toga što se identičan skup zadataka izvršava više puta. Zbog toga je važno da redosled izvršavanja zadataka bude različit za svaki od uslova. Na primer, u slučaju dva zadatka, u prvom uslovu (pretraga pomoću polja i dugmeta) ispitanici obavljaju zadatak 1 pa zadatak 2, dok u drugom uslovu (pretraga kretanjem kroz interfejs) ispitanici obavljaju zadatak 2 pa zadatak 1. Na ovaj način se može ublažiti ili eliminisati efekat učenja.

Varijanta *uparivanja* definiše parove korisnika na osnovu osobina (nezavisnih varijabli) kao što su stručnost i pol. Svaki par je angažovan u svakom uslovu testiranja. Mana ove varijante je da druge nezavisne varijable koje nisu razmatrane mogu uticati na rezultate. Izbor nezavisnih varijabli kao kriterijuma za uparivanje zavisi od konteksta zadatka. Na primer, iskustvo navigacije i pretrage u korisničkim interfejsima može uticati na performanse korisnika. Ova nezavisna varijabla je relevantan kriterijum za uparivanje korisnika u slučaju posmatranog scenarija pretrage interfejsa.

Tabela 5 prikazuje prednosti i mane alternativa eksperimentalnog dizajna.

Tabela 5. Kritička analiza alternativa eksperimentalnog dizajna testiranja upotrebljivosti.

Varijanta	Prednosti	Mane
Između grupa	Nema efekta učenja Lakša za planiranje i analizu Jednostavnije i brže testiranje	Veći broj ispitanika Varijabilnost osobina ispitanika
Unutar grupe	Manji broj ispitanika Nema varijabilnosti osobina ispitanika	Efekat učenja Složenija za planiranje i analizu
Uparivanje	Slično kao između grupa, sa redukovanom varijabilnošću osobina ispitanika	Veći broj relevantnih kriterijuma za uparivanje ispitanika koji mogu uticati na rezultate

8.3.2.4. Statistička analiza rezultata

Podaci prikupljeni testiranjem upotrebljivosti opisuju performanse korisnika u izvršavanju zadataka poput vremena obavljanja zadataka, vremena obavljanja delova zadataka i broj grešaka po zadatku. Analiza podataka poredi performanse ispitanika za različite uslove eksperimenta. Vreme izvršenja zadatka i broj grešaka se agregiraju kao prosečne vrednosti po specifičnim uslovima i porede kako bi se ustanovile razlike. Koriste se standardni statistički testovi, kao što je t-test koji poredi razlike između uslova eksperimenta testiranja kako bi se identifikovale značajne razlike. Na primer, test će okriti da li i u kojoj meri je kontekstni ili kaskadni meni brži u izboru stavki.

Postoji veći broj statističkih instrumenata koji se koriste u oblasti IČR kako bi se analizirale verovatnoće javljanja rezultata. Jedan od najčešće korišćenih je t-test. Rezultati, kao što je vreme za koje je svaki ispitanik izabrao stavke menija u oba uslova (kontekstni i kaskadni meni), se koriste za računanje prosečne vrednosti (*mean*) i standardne devijacije (SD). SD je statistička mera varijabilnosti u odnosu na prosečnu vrednost. T-test koristi jednostavnu formulu koja ispituje značaj razlike između srednjih vrednosti za dva uslova testiranja. Ukoliko je razlika značajna možemo odbaciti nultu hipotezu i prihvatiti alternativnu hipotezu. Na primer, t-test koji poredi vremena izbora stavki menija za dve grupe od 10 i 13 korisnika može dati sledeće rezultate:

$$t=4.53, p<0.05, df=21,$$

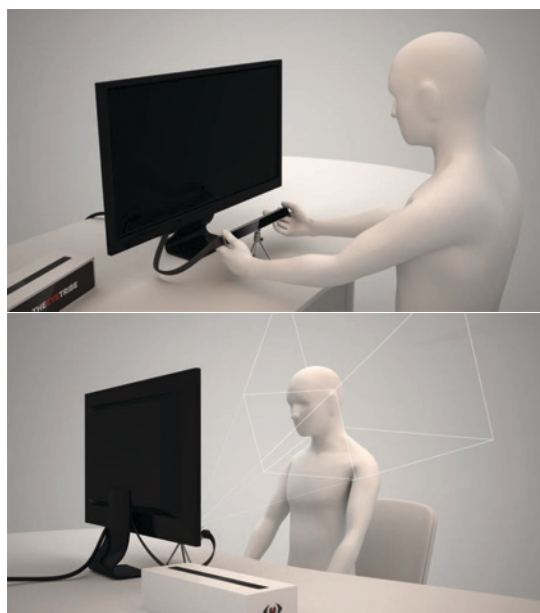
Oznaka *t* je rezultat t-testa, a *df* su stepeni slobode (eng. *degrees of freedom*) kao broj vrednosti u uslovima koje mogu varirati. Računa se po formuli $df =$

$(Na - 1) + (Nb - 1)$, gde je Na broj ispitanika u prvom uslovu (kaskadni meni) i Nb broj ispitanika u drugom uslovu (kontekstni meni). U posmatranom primeru je $df = (10-1)+(13-1) = 21$. Oznaka p je verovatnoća da pronađeni efekat (na primer, razlika u vremenima između dva uslova) nije slučajan. Konkretno, $p < 0.05$ znači da pronađeni efekat nije slučajan, odnosno da je verovatnoća slučajnog nastanka efekta 5%. To znači da postoji razlika u performansama izvršavanja korisnika između dva uslova testiranja. Tipično se vrednost $p < 0.05$ uzima kao dovoljna za odbacivanje nulte hipoteze. Sa druge strane, što je vrednost manja, to je tvrdnja pouzdanija. Na primer, za $p < 0.01$ slučajnost efekta je još manja, odnosno verovatnoća slučajnog nastanka je 1%.

Detaljnija objašnjenja t-testa se mogu naći u [66, 67].

8.4. Praćenje pogleda korisnika i kursora miša

Jedan od najvažnijih povratnih podataka kod testiranja upotrebljivosti sa krajnjim korisnicima jeste pozicija ekrana u koju je korisnik u određenom trenutku gledao. Na osnovu ovih podataka mogu se izvesti različite informacije - koji deo Veb stranice je najzanimljiviji korisnicima, da li se elementi nalaze na očekivanim pozicijama i slično [68]. Dve najpopularnije metode za otkrivanje ovih pozicija su praćenje pogleda korisnika (Slika 87) i praćenje kursora miša.



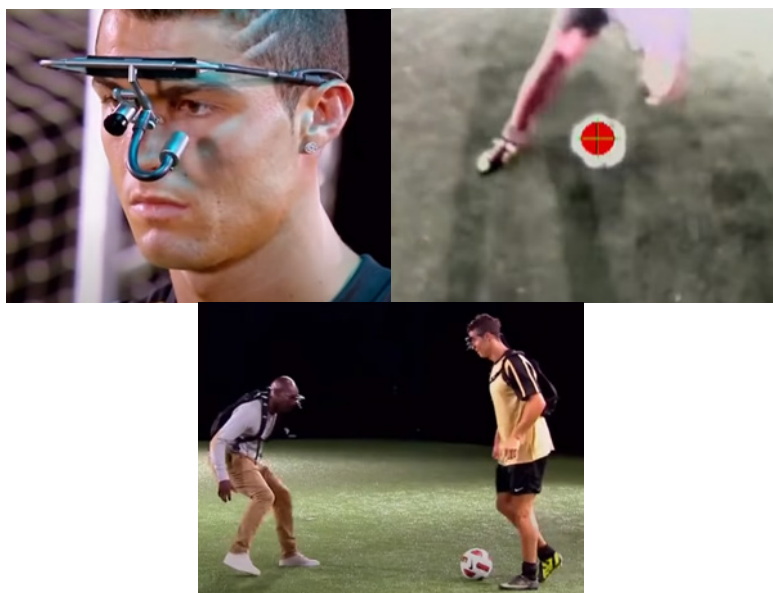
Slika 87. Način korišćenja stonog uređaja za praćenje pogleda *The Eye Tribe*.

Za praćenje pogleda korisnika najčešće se koriste namenski uređaji (engl. *eye tracker*) koji funkciniraju po principu snimanja očiju korisnika i izdvajanja informacije u kom pravcu su usmerene zenice (Slika 88). Ovi uređaji mogu biti stoni ili prenosivi. Stoni uređaji se koriste za snimanje pogleda korisnika kod korišćenja aplikacija putem računarskog ekrana. Obično se kupuju i instaliraju kao nezavisni uređaji, a ponekad dolaze i kao integrisani u računarske monitore. Najčešće je na početku sesije potrebno izvršiti kalibraciju uređaja, odnosno prilagođavanje korisniku, što se čini pomeranjem pogleda na različite, označene delove ekrana. Kvalitet dobijenih podataka u velikoj meri zavisi od pomeranja glave u odnosu na ekran (u nekim slučajevima se koriste i držači za fiksiranje), odnosno od mogućnosti algoritama da otkriju i isprave nastale razlike.



Slika 88. Primer prenosivog uređaja za snimanje pogleda *Tobii Pro Glasses 3*. Naočare (levo) poseduju kamere koje snimaju oči korisnika, kao i kameru koja snima ono što korisnik vidi. Naočare se povezuju sa uređajem (desno) koji snima podatke i obezbeđuje napajanje električnom energijom iz baterije.

Prenosivi uređaji za praćenje pogleda namenjeni su eksperimentima u kojima korisnik ne gleda fiksiranu površinu monitora već se kreće i gleda u različitim pravcima. Najčešće poseduju dve infra-crvene kamere za pojedinačno snimanje očiju, kao i jednu kameru koja snima ono što korisnik vidi. Napajanje el. električnom energijom se obezbeđuje putem baterije, a podaci se snimaju i čuvaju na spoljnoj memoriji, na primer *SD* kartici. Obrada podataka (analiza snimljenih slika očiju i određivanje smera pogleda) najčešće se obavlja lokalno - na drugom računaru, nakon završetka sesije. Prenosivi uređaji za praćenje pogleda mogu imati vrlo visoku preciznost ukoliko se ispravno kalibrišu na početku sesije i tokom nje ne dođe do pomeranja naočara u odnosu na oči korisnika.



Slika 89. Mobilni uređaj za praćenje pogleda korišćen u dokumentarnom filmu *Ronaldo: Tested to the Limit* (2011).

Osim za izvođenje eksperimenata sa krajnjim korisnicima i evaluaciju korisničkog interfejsa, praćenje pogleda korisnika se može koristiti i kao metod za interakciju sa računarskim sistemom. Na primer, informacija o pogledu korisnika se u savremenim video igrama koristi kao kanal interakcije, pa usmeravanje pogleda na određeni objekat (na primer skrivenog neprijatelja) može inicirati njegovu reakciju (na primer napad). Dodatno, praćenje pogleda korisnika može biti i primarni kanal za interakciju korisnika sa računarom, na primer u slučaju kvadriplegije kada korisnik nema mogućnost korišćenja ekstremiteta za kontrolu ulaznih uređaja. U takvim situacijama se kursor miša pomera na osnovu praćenja pogleda korisnika a treptaji interpretiraju kao levi/desni klik.

Značajnu alternativu korišćenju namenskih stonih uređaja za praćenje pogleda predstavlja upotreba Veb kamere personalnih računara. Ova mogućnost se sve češće koristi kod praćenja pogleda kod korišćenja Veb aplikacija, sa obzirom na dobru sistemsku podršku i razvoj kvalitetnih biblioteka za tu namenu. Za razliku od namenskih uređaja, Veb kamere obično ne koriste infra-crveno svetlo, što povećava nivo grešaka, i imaju samo jedan senzor. Iako je kvalitet ovako dobijenih podataka obično znatno niži u odnosi na namenske uređaje, mogućnost izbegavanja korišćenja namenskog uređaja je dragocena, prvenstveno zato što nije neophodno da se korisnici nalaze u laboratoriji, već se snimanje može obaviti sa bilo kog računara koji ima instaliranu Veb kameru.



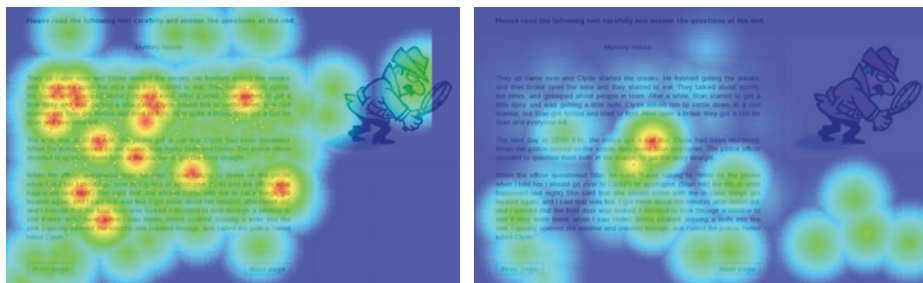
Slika 90. *WebGazer.js* biblioteka nudi mogućnost korišćenja Veb kamere za praćenje pogleda korisnika koji pristupaju Veb stranici.

Praćenje pozicija na kojima se nalazi kursor miša (eng. *mousetracking*) predstavlja često korišćenu alternativu praćenju pogleda korisnika. Osnova za primenu ove metode je postojanje visoke korelacije između pozicije na kojoj se nalazi pogled korisnika i pozicije na kojoj se u tom trenutku nalazi kursor miša. Kod ove metode poseban softver ili softverski modul beleži u kom trenutku se kursor miša nalazio na kojoj poziciji (x,y koordinate). Koordinate mogu biti relativne (odnosno određene u odnosu na područje ekrana koje zauzima program koji se koristi) ili apsolutne (u odnosu na ceo ekran). Vreme na koje se izmerena pozicija odnosi najčešće se meri sa preciznošću u milisekundama. Ova metoda često podrazumeva i uporedno, najčešće periodično snimanje izgleda ekrana, kako bi se omogućila naknadna reprodukcija snimljene sesije [69].



Slika 91. Primer toplotne karte (eng. *heatmap*) dobijene praćenjem kursora miša posetilaca sajta racunarskemreze.com.

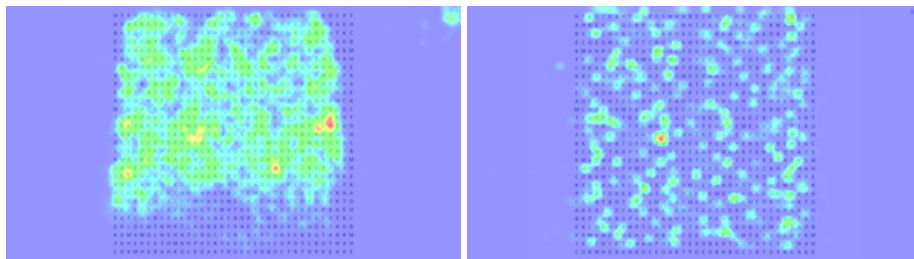
U odnosu na praćenje pogleda korisnika, praćenje kursora miša nudi nekoliko značajnih prednosti. Kao prvo, nije potrebno posedovanje posebnog uređaja za praćenje pogleda, već je dovoljno korišćenje odgovarajućeg softvera. Zatim, nije potrebno vršiti kalibraciju uređaja na početku sesije. Konačno, a možda i najznačajnija prednost ovog metoda je mogućnost korišćenja mimo laboratorijskih uslova, na primer na Veb sajtu koji posećuju korisnici sa različitih fizičkih lokacija.



Slika 92. Toplotne karte pogleda korisnika (levo) i kursora miša (desno) za istu sesiju (srednje/nisko-interaktivni sadržaj).

Sa druge strane, postoje i značajni nedostaci praćenja kursora miša u odnosu na pogled korisnika. Glavni nedostatak jeste nepreciznost samog metoda, odnosno mogućnost da se javi značajna razlika između pozicija na kojima se nalaze

pogled i kursor. Ova razlika je uglavnom manje izražena kod visoko-interaktivnih sadržaja (na primer Veb aplikacije sa intenzivnim angažovanjem korisnika) dok može sam metod učiniti potpuno neupotrebljivim u slučaju sadržaja sa niskom interaktivnošću (na primer gledanje video zapisa).



Slika 93. Viša korelacija između toplotnih karata pogleda korisnika (levo) i kursora miša (desno) u slučaju visoko-interaktivne aplikacije.

Dva najčešća oblika reprezentacije podataka dobijenih praćenjem pogleda korisnika ili kursora miša jesu animacije i toplotne karte. Animacije predstavljaju potpunu vizuelnu rekonstrukciju sesije, koja uključuje i vremensku dimenziju. Nedostatak ovog tipa vizuelizacije predstavlja veliko vreme potrebno za analizu rezultata i subjektivnost kod sumiranja snimljenog materijala. Nasuprot tome, toplotne karte predstavljaju vizuelnu reprezentaciju kod koje je vremenska dimenzija prevedena u intenzitet boje. Na primer, pozicije na kojima je pogled/kursor proveo dosta vremena često se predstavljaju toplim bojama (žuta, narandžasta, crvena), dok se za pozicije sa relativno manjim vremenskim sumama koriste hladnije boje (od zelene ka plavoj boji). Nedostatak toplotnih karata je nepostojanje informacije o vremenskom redosledu aktivnosti.

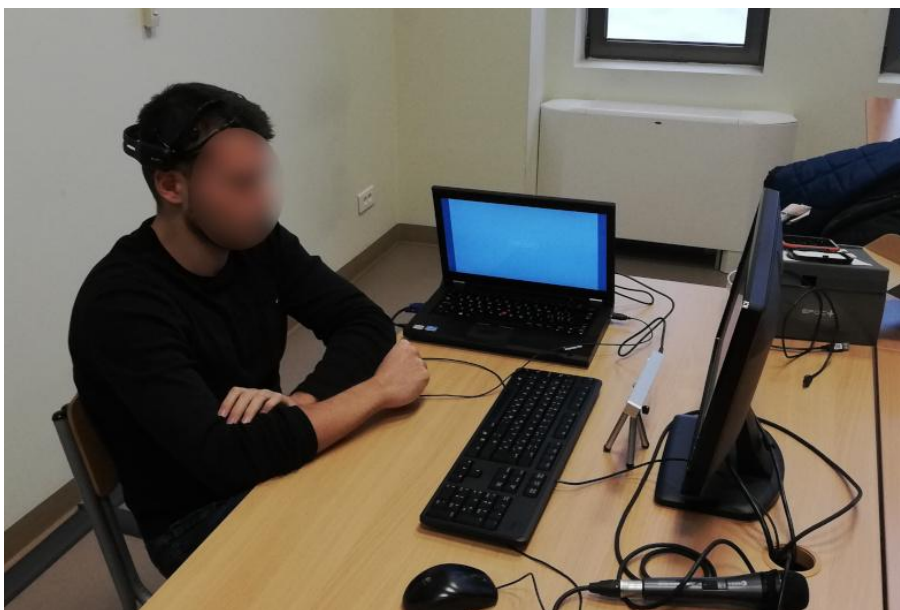


Slika 94. Primer vizuelne reprezentacije klikova.

Softver za praćenje kursora miša najčešće poseduje mogućnost detekcije i čuvanja ostalih događaja GKI, kao što su klik levim/desnim tasterom, skrolovanje i drugi. Još jedna značajna informacija koja se može dobiti na osnovu podataka je na koje pozicije/elemente je korisnik kliknuo. Ova informacija se može koristiti u različite svrhe - na primer utvrđivanje koja je od višestrukih kontrola iste namene najpopularnija kod korisnika. Takođe, njome se mogu identifikovati i greške u dizajnu. Na primer, autori ovog udžbenika su ovom metodom otkrili grešku u dizajnu Veb sajta usled koje nije kompletna površina stavki u meniju reagovala na klik.

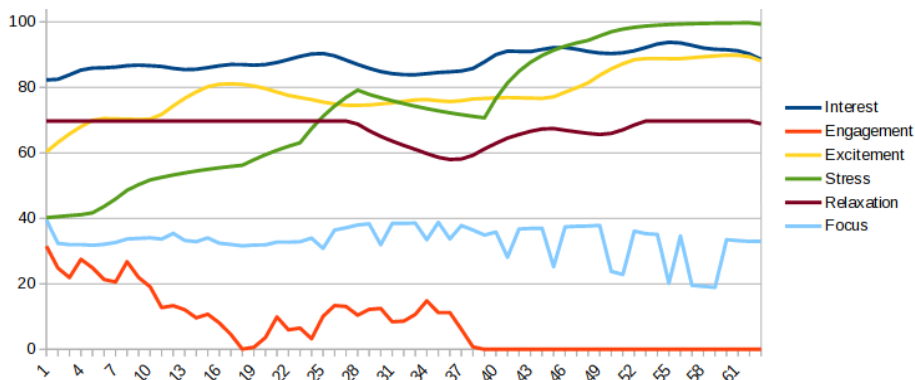
8.5. Primena elektroencefalografije

Elektroencefalografija (eng. *Electroencephalography*, *EEG*) je neurofiziološka metoda za registrovanje i merenje električne moždane aktivnosti. Iako se ova metoda prvenstveno koristi u medicini (na primer za otkrivanje epilepsije, efekata anestezije i slično), u poslednje vreme se pojavljuju i komercijalni uređaji koji mogu da se koriste za analizu interakcije između čoveka i računara. U tom kontekstu se prvenstveno koriste za prikupljanje podataka o mentalnom stanju ispitanika tokom primarne aktivnosti, na primer evaluacije korisničkog interfejsa aplikacije ili Veb sajta.



Slika 95. Primer korišćenja *EEG* i *eye tracker* uređaja u eksperimentima vezanim za interakciju korisnika i računara.

Većoj popularnosti EEG uređaja u IČR doprineo je i razvoj algoritama koji eliminišu potrebu za visokim medicinskim znanjem u cilju interpretacije dobijenih rezultata. Komercijalno dostupni uređaji obično dolaze sa algoritmima za interpretaciju ugrađenim u sam uređaj ili u prateći softver. Ovi algoritmi iz izvornih podataka sa elektroda izračunavaju apstraktnije karakteristike mentalnog stanja korisnika kao što su stres, fokus, uzbuđenost i drugih. Na osnovu ovih podataka može se raditi dodatna evaluacija efektivnosti korisničkog interfejsa, odnosno mogu se otkriti situacije koje podižu stres korisnika, smanjuju interesovanje, i slično [70].



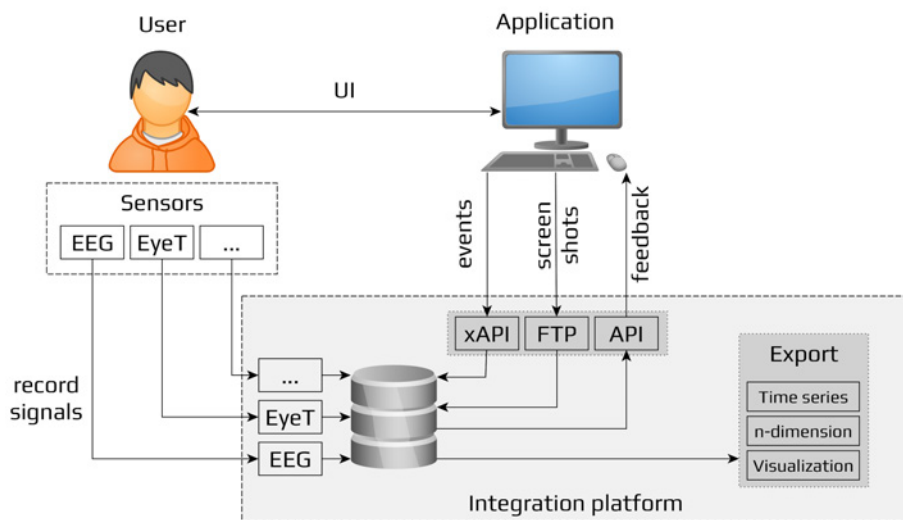
Slika 96. Primer dijagrama izvedenih karakteristika mentalnog stanja korisnika na osnovu podataka dobijenih putem EEG senzora u eksperimentu trajanja 30 sekundi sa učestalošću uzorkovanja od 2Hz.

Razvojem oblasti - primene elektroencefalografije za interakciju sa korisnikom - pojavili su se novi, komercijalno i široko dostupni uređaji sa integrisanim EEG senzorima za obavljanje specifične funkcije. Na primer, uređaji sa integrisanim slušalicama i EEG senzorima, u cilju realizacije navodene meditacije koja se prilagođava mentalnom stanju korisnika.

8.6. Integracija podataka

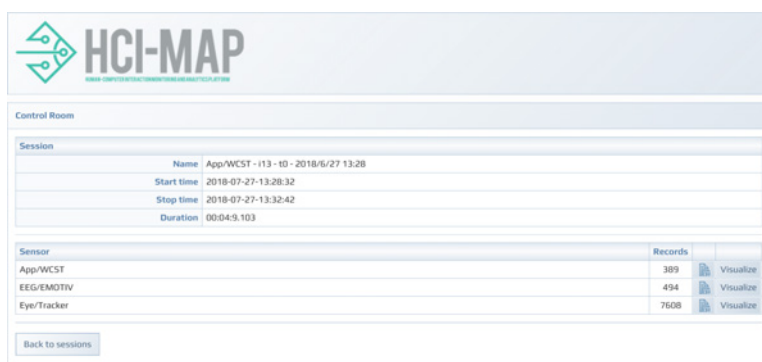
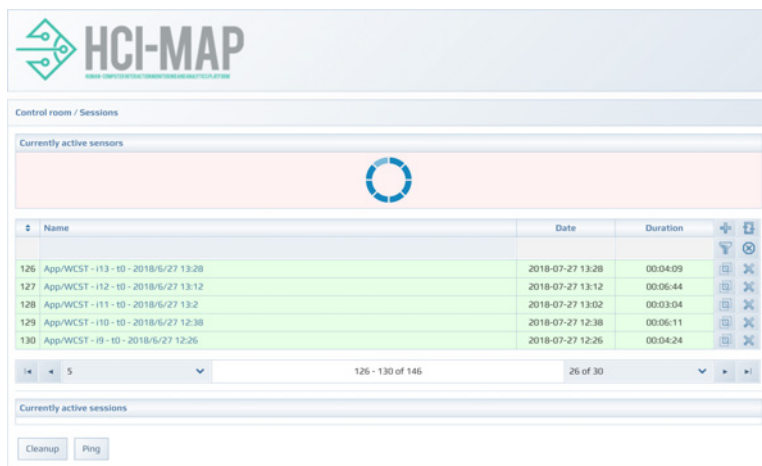
Kvalitet testiranja razvijenog rešenja sa krajnjim korisnicima u velikoj meri zavisi od ispravno organizovanih eksperimenata, kao i od podataka o korišćenju koji se uzimaju u obzir. Na primer, primenom metode praćenja pogleda korisnika ili kursora miša može se pravilnije izvesti zaključak da li je pauza u korišćenju nastala zbog loše organizovanog korisničkog interfejsa ili iz nekog drugog razloga. Ili, uzimanje u obzir arterijskog pulsa korisnika ili primena elektroencefalografije mogu dati informaciju da li je korišćenje interfejsa stresno za korisnika.

Jedan od glavnih izazova u implementaciji/razvoju platforme koja se koristi za izvođenje eksperimenata sa krajnjim korisnicima jeste integracija podataka. Drugim rečima, potrebno je objediniti podatke o korišćenju rešenja koje se evaluira sa podacima dobijenim iz različitih izvora i sa različitih senzora. Slika 97 prikazuje arhitekturu platforme koja se koristi za ovakva testiranja - u pitanju je platforma pod nazivom *Human-Computer Interaction Monitoring and Analytics Platform*, *HCI.MAP*, koju su za potrebe sopstvenih istraživanja razvili autori ovog udžbenika [71].



Slika 97. Arhitektura HCI.MAP platforme za izvođenje eksperimenata sa krajnjim korisnicima.

Prikazana platforma podržava prikupljanje podataka sa različitih senzora, na primer EEG uređaja, *eye tracker*-a, merača arterijskog pulsa, softvera za logovanje aktivnosti miša i tastature, i drugih. Sa druge strane, platforma prikuplja podatke i o korišćenju aplikacije, bilo kroz snimanje ekrana i zvuka, bilo kroz namenski *API* kojim se beleže događaji od interesa u aplikaciji (na primer klik dugmeta za registraciju na Veb sajtu). Sa obzirom na modularnu i generičku arhitekturu platforme, skup podržanih senzora i aplikacija lako je proširiv, a same komponente mogu da koriste različite protokole za komunikaciju sa platformom (Slika 98). Podrazumevani protokol je *HTTP* sa *JSON* kodovanjem podataka ili *xAPI* sa obzirom da se platforma često koristi u oblasti edukacije. Prikupljeni podaci mogu se posle završetka eksportovati, za obradu u drugim softverskim alatima, ili u realnom vremenu obrađivati i prosleđivati na određeni *API* (na primer, za razvoj adaptivnog korisničkog interfejsa koji uzima u obzir stanje korisnika).



Slika 98. Kontrolni panel HCI.MAP platforme sa prikazom sesija, senzora i izvozom podataka.

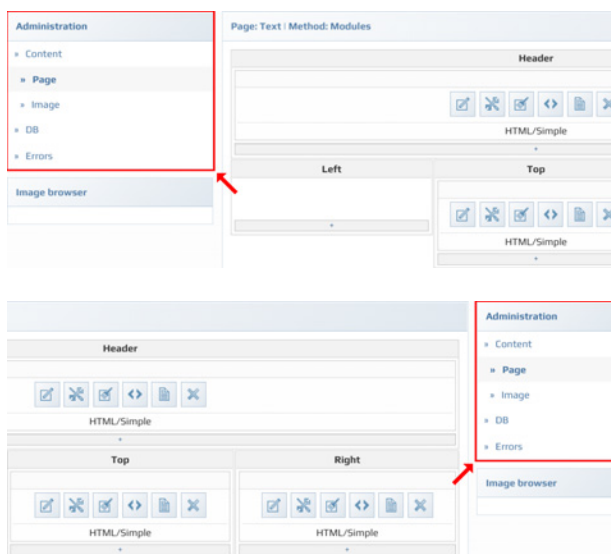
Jedan od glavnih izazova prilikom prikupljanja podataka tokom izvođenja eksperimenta jeste i vremenska sinhronizacija podataka. Sa obzirom na moguću brzinu promena stanja na senzorima ili događaja u interakciji, dozvoljena tolerancija greške svodi se na red milisekundi. Pošto se podaci prikupljaju pomoću većeg broja uređaja (na primer server zadužen za prijem i skladištenje podataka, radna stanica na kojoj se interaguje sa softverom koji se evaluira i uređaji koji kontrolišu određene senzore) neophodno je da interni časovnici svih uređaja budu sinhronizovani. Alternativno rešenje je da se koristi vreme prijema podataka na strani servera, ali je u tom slučaju potrebno optimizovati infrastrukturu tako da ukupna kašnjenja (prenos preko mreže, sistemsko procesiranje, aplikativna obrada) ne prelaze definisanu toleranciju. Za potrebe prethodno prikazane platforme autori su razvili protokol aplikativnog nivoa,

odnosno protokol koji radi unutar *HTTP*-a, koji koristi principe *Network Time Protocol*-a za sinhronizaciju časovnika putem mreže.

8.7. A/B testiranje

Prilikom dizajniranja korisničkih interfejsa često se javljaju situacije u kojima nije jasno kako izabrati optimalno rešenje, odnosno da li će planirana izmena biti efikasna i svrsishodna, bez obzira na prethodno iskustvo i poznavanje principa dizajna. Na primer, za dizajniranje dugmeta na Veb stranici, koje treba da privuče pažnju korisnika, teorija upućuje na izbor crvene boje. Sa druge strane, iskustvo korisnika sa crvenom bojom na semaforima i saobraćajnim znacima suprotstavljeno je željenoj asocijaciji - pokretanju određene akcije, na primer pokretanje procedure za otvaranje novog korisničkog naloga. U takvim situacijama često jedini način da se zaključi koje rešenje je optimalno predstavlja praktična provera i poređenje svih opcija.

A/B testiranje je metod za praktičnu proveru efikasnosti više varijanti potencijalnog rešenja. Krajnjim korisnicima se prezentuju obe (ili sve, u slučaju postojanja više od dve) predložene varijante rešenja. Zatim se statistički meri koja varijanta je dala bolje rezultate, odnosno koja je dovela do željenog rezultata (kraće vreme korišćenja, kupovina proizvoda, registracija korisničkog naloga, i drugi).



Slika 99. Da li je efikasnije postaviti glavni meni sa leve (A) ili sa desne strane (B) ? A/B testiranje može statistički utvrditi koja opcija je efikasnija za korisnike.

Kod A/B testiranja treba razmotriti faktore koji utiču na validnost rezultata. Kao prvo, ispitivanje treba izvršiti na uzorku korisnika dovoljne veličine, kako bi zaključak bio statistički značajan. Zatim, karakteristike ispitanika u kontrolnoj i eksperimentalnoj grupi, kao i uslovi testiranja, treba da budu istovetni, odnosno slični. Ukoliko se uvodi izmena u dizajnu, često je dobro ići obazrivo sa testiranjem, recimo prikazivanjem rešenja koje se testira 20% populacije. Osim ukoliko ima jasnih razloga za takav pristup, treba izbegavati mešanje različitih rešenja tokom jedne sesije korišćenja. Takođe, treba imati u vidu da prikazivanje novog rešenja korisnicima koji su se već navikli na postojeće može uticati na izmerene rezultate.

Česta greška kod A/B testiranja jeste i istovremeno testiranje više promenljivih. Na primer, kombinovanje pozicije menija sa bojom dugmeta za akciju. Ukoliko ove promenljive nisu međusobno zavisne, može se desiti da se njihovim istovremenim testiranjem dobiju nejasni ili čak netačni rezultati.

Treba imati u vidu da A/B testiranje nema mogućnost da uputi na optimalno rešenje, već samo da uporedi efikasnost predloženih opcija. Dakle, A/B eksperimenti kojim se testira efikasnost bele i crne pozadine korisničkog interfejsa nemaju mogućnost da otkriju na primer plavu boju kao optimalnu. Takođe, i redosled upoređivanja opcija može imati uticaja na rezultate. Iz tih razloga se često A/B eksperimenti izvode u velikom broju iteracija, a veličina eksperimentalne grupe određuje inverzno u odnosu na uticaj koji predložena izmena može imati.

9. Heuristička evaluacija

Do sada opisane vrste testiranja podrazumevaju učešće krajnjih korisnika kao ispitanika. Postoje i metode zasnovane na opštim znanjima o korisnicima, sadržanim u principima koji se nazivaju heuristike upotrebljivosti ili matematičkim modelima koje opisuju karakteristično ponašanje korisnika (kao što je Fitsov zakon). Ove metode ne zahtevaju prisustvo korisnika u toku testiranja. Heuristička evaluacija je najzastupljeniji metod testiranja bez prisustva krajnjih korisnika. Pored Fitsovog zakona, postoje i drugi modeli čiji se pregled može naći u [72].

Ponekada korisnici iz ciljne grupe nisu dostupni za testiranje, ili je njihovo regrutovanje i angažovanje skupo i složeno. U takvim slučajevima, drugi korisnici mogu učestvovati u testiranju i dati povratnu informaciju u vezi sa korišćenjem interfejsa. Ovi korisnici poseduju iskustvo, znanja i veštine u vezi sa dizajnom interakcije i tipičnim ponašanjima korisnika. Poznati su kao eksperti ili inženjeri upotrebljivosti (eng. *usability experts or engineers*). U praksi se umesto eksperata mogu angažovati i neutralni korisnici koji nisu eksperti u datim oblastima.

Metod je razvijen 90-tih godina 20. veka kao alternativa testiranju upotrebljivosti sa krajnjim korisnicima. Njegov nastanak je inspirisan i automatizovanim metodama za testiranje softvera koje su se koristile u to vreme. Vremenom, kroz praksu, metod je unapređivan kako bi se mogao primenjivati za procenu upotrebljivosti različitih vrsta GKI.

9.1. Heuristike upotrebljivosti

Heurističku evaluaciju kao analitički metod testiranja (bez krajnjih korisnika) su originalno razvili Jakob Nilsen i Rolf Molik [73]. Metod je razvijen empirijski, analizom 249 problema upotrebljivosti. Nakon toga je dalje razvijan [36, 56] i prilagođavan za korisničke interfejse specifičnih aplikacija kao što su ambijentalni sistemi [74], video igre [75] i virtuelna stvarnost [76]. U toku evaluacije eksperti upotrebljivosti procenjuju da li su elementi korisničkog interfejsa poput dijaloga, menija, dugmadi, ikona, rasporeda, navigacione strukture, pomoći, i drugih u skladu sa heuristikama. Heuristike su formulisane u stilu opštih principa dizajna, onako kako ih vide krajnji korisnici kao što su konzistentnost u dizajnu, smanjenje kognitivnog naprezanja, i doživljaj kontrole nad interfejsom. Na taj način ih može razumeti čak i osoba koja nije ekspert i koristiti za vrednovanje upotrebljivosti interfejsa. Koncipirane su tako da u isto vreme predstavljaju kriterijume zadovoljenja atributa upotrebljivosti (analitička uloga) i univerzalne smernice za dizajn upotrebljivog interfejsa (sintetička i kreativna uloga).

Heuristike upotrebljivosti korisničkog interfejsa su definisane kao [73]:

- **H1: Vidljivost statusa sistema** (eng. *visibility of system status*) - Sistem treba da informiše korisnika o tome šta se dešava u toku korišćenja kroz odgovarajuću i pravovremenu povratnu informaciju.
- **H2: Usaglašenost sistema sa realnim svetom** (eng. *match between system and the real world*) - Sistem treba da govori jezikom korisnika, korišćenjem reči, fraza i koncepata poznatim korisniku, a ne tehničkih termina. Sistem treba da prati konvencije iz realnog sveta tako da se informacija pojavljuje u prirodnom i logičnom redosledu.
- **H3: Kontrola i sloboda u korišćenju** (eng. *user control and freedom*) - Korisnik ima kontrolu nad korišćenjem sistema, a ne obrnuto. U slučaju da izabere funkciju greškom, mora mu se ponuditi jasno označen izlaz iz kritične situacije. Podržati reverzibilnost akcija (eng. *undo and redo*).
- **H4: Konzistentnost i standardi** (eng. *consistency and standards*) - Identične reči, situacije i akcije (simboli) moraju imati identično značenje. Različite reči, situacije i akcije (simboli) moraju imati različito značenje.
- **H5: Prevencija grešaka** (eng. *error prevention*) - Dizajn mora voditi računa o prevenciji grešaka pre nego one nastanu. Eliminirati uslove i situacije koji mogu izazvati grešku, i tražiti potvrdu pre izvršenja kritične akcije.
- **H6: Prepoznavanje pre pamćenja** (eng. *recognition rather than recall*) - Minimizovati mentalno naprezanje korisnika i učiniti objekte, akcije i opcije jasno vidljivim.
- **H7: Fleksibilnost i efikasnost korišćenja** (eng. *flexibility and efficiency of use*) - Sistem se mora prilagoditi korisnicima bez iskustva korišćenja i korisnicima sa iskustvom korišćenja.
- **H8: Estetski i minimalistički dizajn** (eng. *aesthetic and minimalistic design*) - Interfejs ne treba da sadrži informaciju koja nije relevantna ili se retko koristi. Svaka nova informacija privlači dodatnu pažnju korisnika i može skrenuti fokus sa bitnih informacija.
- **H9: Prepoznavanje i oporavak od grešaka** (eng. *help users recognize, diagnose, and recover from errors*) - Poruke o greškama moraju biti napisane jezikom korisnika (bez kodova), precizno opisati problem i predložiti rešenje problema.
- **H10: Pomoć i dokumentacija** (eng. *help and documentation*) - Iako je poželjno da se sistem može koristiti bez dokumentacije, ona može biti potrebna. Dokumentacija mora biti laka za pretragu, povezana sa zadatkom korisnika, konkretna, i ne previše obimna.

U Poglavlju 3 pojam upotrebljivosti je detaljnije opisan pomoću odgovarajućih atributa. Svaka heuristika unapređuje upotrebljivost fokusom na specifičan atribut.

Tabela 6 daje vezu heuristika upotrebljivosti sa atributima koje u najvećoj meri podržavaju.

Tabela 6. Primarno podudaranje heuristika i atributa upotrebljivosti.

Heuristika upotrebljivosti	Atribut upotrebljivosti
H1: Vidljivost statusa sistema	Greške
H2: Usaglašenost sistema sa realnim svetom	Lakoća učenja
H3: Kontrola i sloboda u korišćenju	Greške
H4: Konzistentnost i standardi	Lakoća učenja
H5: Prevencija grešaka	Greške
H6: Prepoznavanje pre pamćenja	Greške
H7: Fleksibilnost i efikasnost korišćenja	Efikasnost
H8: Estetski i minimalistički dizajn	Lakoća učenja
H9: Prepoznavanje i oporavak od grešaka	Greške
H10: Pomoć i dokumentacija	Lakoća učenja

Elementi korisničkog interfejsa se procenjuju u odnosu na heuristike upotrebljivosti. Na primer, u slučaju aplikacije društvenih mreža evaluator može procenjivati kako bi korisnik kreirao nove kontakte. Evaluator analizira interfejs, ispituje funkcije specifičnih elemenata, i upoređuje sa heuristikama.

Važnost heuristika varira u zavisnosti od vrste i namene korisničkog interfejsa. Relevantnost heuristika u datom kontekstu je određena atributima upotrebljivosti koji imaju viši prioritet. Broj korisnika za sprovođenje heurističke evaluacije je identičan kao kod testiranja upotrebljivosti sa krajnjim korisnicima (Poglavlje 8.3.1). Veći broj evaluatora će dati bolje rezultate, ali je i skuplji za angažovanje. Angažovanje korisnika sa odgovarajućim iskustvom i nepostojanje zahteva u vezi sa posebnim uslovima i opremom čini ovaj tip evaluacije popularnima među dizajnerima i inženjerima. Zbog toga se naziva i jeftinom evaluacijom (eng. *discount evaluation*). Za brzu procenu interfejsa rano u razvoju, jedan do dva korisnika mogu pronaći veći deo problema upotrebljivosti. Na kraju procesa razvoja se može angažovati veći broj korisnika za sumativnu heurističku evaluaciju interfejsa.

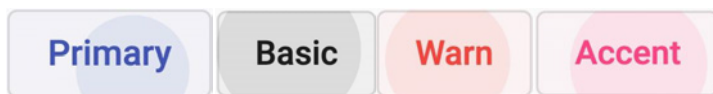
Originalne heuristike upotrebljivosti su univerzalne, primenjive na GKI iz različitih domena. Pored toga, u praksi su heuristike prilagođavane specifičnim vrstama interfejsa. Izvedene heuristike se oslanjaju na originalne, i istovremeno naglašavaju aspekte korisničkog interfejsa koji si relevantniji za dati tip aplikacije. Na primer, Endi Bad je predložio skup heuristika za Veb sajtove u kojem je akcenat na kvalitetu informacija interfejsa [77], dok se heuristike za mobilne interfejse fokusiraju na kvalitet informacija i navigaciju [78].

U nastavku će svaka heuristika biti detaljnije opisana.

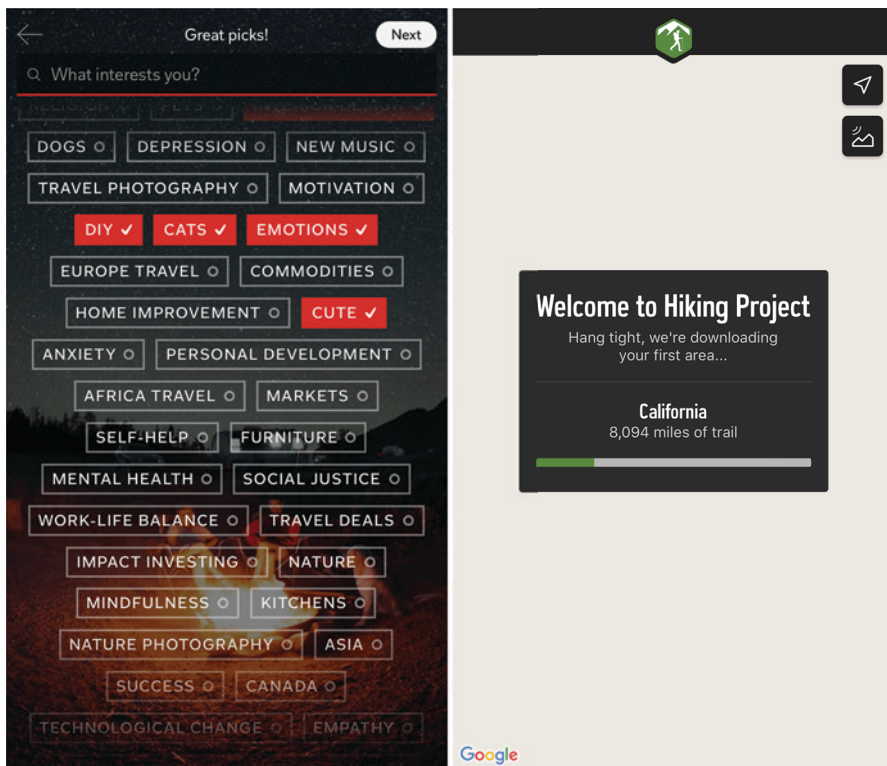
9.1.1. H1: Vidljivost statusa sistema

Pravilo govori da je jedan od preduslova upotrebljivosti da korisnički interfejs obaveštava korisnika u toku korišćenja interfejsa. Ova obaveštenja su diskretna, tako da ne ometaju tok misli i akcija korisnika, već mu pomažu time što daju informacije o rezultatima njegovih akcija kroz prikladnu i pravovremenu povratnu informaciju. Prikladna znači da se informacija odnosi na tekuću, neposredno završenu ili predstojeću akciju korisnika. Pravovremena znači da se povratna informacija mora pojaviti u vreme izvršenja akcije. Akcija je povezana sa pokretanjem ili izborom konkretnog elementa korisničkog interfejsa (dugme, ikona, stavka menija, link i drugi). Pravilna primena ima dve važne posledica. Pre svega, smanjuje se verovatnoća da će korisnik naćiniti grešku time što se obezbeđuje neka vrsta dokaza ispravnosti izbora opcija koje korisnik vrši na interfejsu. Pored toga, ovo unapređuje interfejs u pogledu lakoće ućenja (tako što utiće na izbor ispravne opcije) i povećava efikasnost korišćenja interfejsa (proces donošenja odluka prilikom korišćenja interfejsa ćini bržim). Još jedna posledica je da korisnik razvija psihološki doživljaj kontrole nad sistemom (time što smanjuju doživljaj nesigurnosti kod korisnika) i razvija poverenje u sistem. Ovo je važan faktor prihvatanja tehnologije [8].

Slika 100a daje primer povratne informacije o događaju klika na dugme kao promenu izgleda elementa kroz takozvani vizuelni efekat krućnog talasa. Slika 100b ilustruje povratnu informaciju kao vizuelno isticanje (eng. *highlighting*) izbora korisnika, i indikator progresa operacije ćije izvršavanje traje odrećeno vreme.



(a)



(b)

Slika 100. Vidljivost statusa sistema - (a) Promena izgleda dugmeta prilikom klika; (b) Promena boje i dodavanje simbola (✓) korisniku poručuje da je sistem registrovao izbore korisnika (levo). Indikator progressa govori korisniku da operacija sistema zahteva određeno vreme (desno).

Izvor: <https://www.nngroup.com/articles/visibility-system-status/>.

9.1.2. H2: Usaglašenost sistema sa realnim svetom

Ova načelo polazi od pretpostavke da se korisnik oseća komforno kada je familijaran sa objektom korišćenja. Drugim rečima, prilikom upotrebe interfejsa se oslanjamo na iskustvo iz korišćenja objekata iz realnog i digitalnog sveta.

U kontekstu razvoja korisničkog interfejsa, ovo opšte načelo se može konkretizovati kroz specifične elemente korisničkih interfejsa. Ovde se govori o

načinima na koji ovi elementi svojim izgledom i ponašanjem saopštavaju informacije korisniku.

Terminologija, odnosno jezik interfejsa mora biti razumljiv za korisnike. Ako korisnici ne razumeju terminologiju interfejsa to će verovatno uzrokovati probleme u interpretaciji servisa koje interfejs pruža. Ovo može otežati učenje korišćenja interfejsa. Korisnik mora da razume značenje termina bez korišćenja pretrage ili rečnika. Slika 101a daje dobar primer jezika korisničkog interfejsa koji ne koristi tehničke termine ili oznake koje ciljna grupa korisnika možda neće razumeti. Sa druge strane, Slika 101b daje primer interfejsa koji koristi stručne termine i izraze iz različitih oblasti koji korisnicima mogu biti nepoznati i nerazumljivi.

PREVENTIVE CARE IS IMPORTANT TO YOUR HEALTH

Health Benefits & Coverage

WHAT IS PREVENTIVE CARE?
Preventive care (sometimes called "preventative care") is routine health care that includes screenings, services and counseling to help prevent illness, disease or other health problems. It's covered by your plan because it can help you stay healthy and lets you know if you may develop a health problem.

UNDERSTANDING YOUR PREVENTIVE CARE BENEFITS
As mentioned in your benefits booklet, many screenings and tests are covered at 100% with no out-of-pocket costs to you as part of your health plan. Use your preventive care and stop health problems before they get serious.

Don't let disease sneak up on you. Preventive care could save your life.

[SUMMARY OF PREVENTIVE CARE SERVICES](#)

WHICH TYPES OF SERVICES ARE COVERED AT 100%?

Preventive care that is 100% covered includes many important services like the ones listed below. This list is not complete, so please see the full list of preventive care services and any limitations on the [Men](#), [Women](#) and [Children](#) pages.

(a)

Abacus Private Cloud Quick Facts

- 8 Data Centers
- 256-bit AES Data Encryption
- No Fee for After Hours Support
- Accessible Anywhere, Anytime
- 100% Renewable Data Center Power
- 99.999% Uptime Service Level Agreement
- No Fee for Client Data Export
- Up to 60 Months Price Guarantee
- SOC1/SOC2/SOC3/SSAE16 Certifications
- Unlimited Cloud Platform Support
- No Fee for Device Reconfiguration

Law Firm Business Continuity Management

A cloud-based legal software solution, Abacus Private Cloud offers fully integrated backup and disaster recovery cloud technology in addition to custom Business Continuity solutions that address both catastrophic and clinical events to protect your firm's confidential and irreplaceable case and client data from the unpredictable.

[Learn More](#)

Full-Spectrum Security

Abacus Private Cloud is complete with robust security and compliance policies, including two-factor authentication, five tiers of physical, network and data security, information handling and data privacy protocols in accordance with NIST standards, and geographically redundant, SSAE-16 and SOC-1, 2, and -3 compliant data centers.

[Learn More](#)

Zero Capital Investment

Private Cloud environments reduce up-front capital expenses such as server hardware and software, backup hardware and software, network infrastructure and installation fees. Leverage your operating expenses budget while speeding time to deployment with Abacus Private Cloud virtualization technology.

Complete Ownership of Data

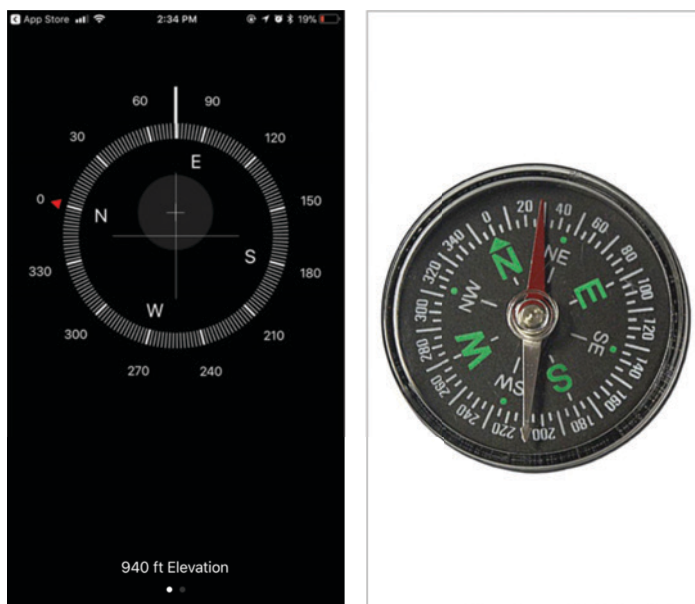
Unlike other Public Cloud storage and SaaS providers that may keep you from accessing your own data, with Abacus Private Cloud you own your data. Upon termination of any business relationship, you will always have access to your irreplaceable case and client data.

(b)

Slika 101. Usaglašenost sistema sa realnim svetom - Jezik korisničkog interfejsa mora biti razumljiv krajnjim korisnicima.

Izvor: <https://www.nngroup.com/articles/match-system-real-world/>.

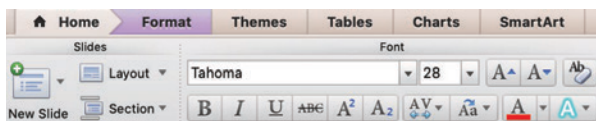
Sledeće sredstvo dizajna korisničkog interfejsa su metafore kao analogije sa aktivnostima i objektima iz realnog i digitalnog sveta. Interakcija koja podseća na fizički svet se oslanja na postojeće znanje korisnika i doprinosi lakšem učenju korišćenja interfejsa. Korisnik formira mentalni model o tome kako sistem funkcioniše na osnovu prethodnog iskustva. Nakon toga, implicitno se oslanja na formirani model kada se susreće sa sličnim digitalnim sistemom. Sličnost se odnosi na izgled i ponašanje između elemenata naučenog i novog sistema. Na ovaj način, metafore mogu pospešiti primenu mentalnog modela. Slika 102 prikazuje digitalnu formu fizičkog uređaja kao primer metafore na nivou čitave aplikacije.



Slika 102. Usaglašenost sistema sa realnim svetom - Fizički kompas (desno) kao metafora za virtuelni kompas (levo).

Izvor: <https://www.nngroup.com/articles/match-system-real-world/>.

Neki od elemenata GKI koje često koristimo su zapravo metafore. Na primer, kartica element (eng. *tab*) za organizovanje sadržaja korisničkog interfejsa je zasnovan na fasciklama sa listićima radi lakšeg pronalaženja sadržaja. Klizač kao element korisničkog interfejsa je nastao po uzoru na ručice kao kontrole u audio uređajima.



Slika 103. Usaglašenost sistema sa realnim svetom - Metafore se mogu naći i u standardnim elementima GKI kao što je kartica.

Metafore imaju šire značenje i ne odnose se isključivo na pojedinačne objekte i kontrole interfejsa, već i na celokupan izgled koji utiče na doživljaj interfejsa. Na primer, isticanje delova članaka na platformi *Medium* asocira na korišćenje fizičkih markera za označavanje teksta.

Chatbots could also use different greetings instead of the boring 'Hi' every time in response to a chat query. Something like 'Hey, How are you doing today?' or even a 'Hello, what's new with you?' could add another dimension and connect with the chatbot.

We need to make sure that a bot's behavior and personality should be consistent throughout not just in one instance of the conversation but through every interaction with the end user. We don't want a bot with multiple personalities showing up because of inconsistent responses to chat queries.



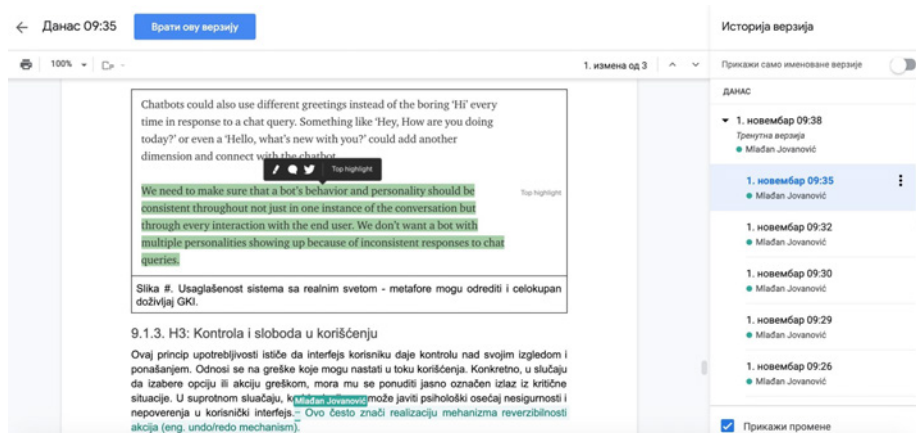
Top highlight

Slika 104. Usaglašenost sistema sa realnim svetom - Metafore mogu odrediti i celokupan doživljaj GKI.

9.1.3. H3: Kontrola i sloboda u korišćenju

Ovaj princip upotrebljivosti ističe da interfejs korisniku daje kontrolu nad svojim izgledom i ponašanjem. Odnosi se na greške koje mogu nastati u toku korišćenja. Konkretno, u slučaju da izabere opciju ili akciju greškom, mora mu se ponuditi jasno označen izlaz iz kritične situacije. U suprotnom slučaju, kod korisnika se može javiti psihološki doživljaj nesigurnosti i nepoverenja u korisnički interfejs. Ovo često znači realizaciju mehanizma reverzibilnosti akcija. Ovde se ne misli samo na grešku kao izbor pogrešne opcije interfejsa, već i na prethodne aktivnosti koje korisnik želi da promeni ili koriguje. Na ovaj

način se sistem vraća u stanje pre nastanka greške ili od kojeg je moguće izvršiti željene promene.

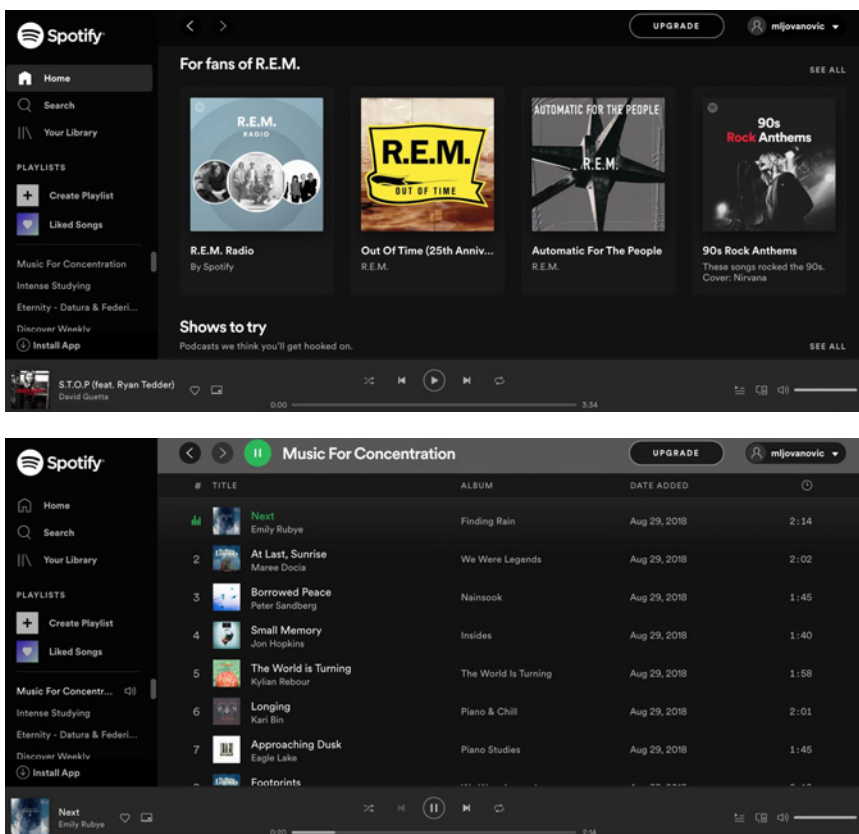


Slika 105. Kontrola i sloboda u korišćenju - Aplikacija Gugl dokumenti (eng. *Google docs*) prati istorijat rada korisnika i daje hronološki spisak izmena kako bi se sadržaj mogao menjati od trenutka prošlog vremena.

9.1.4. H4: Konzistentnost i standardi

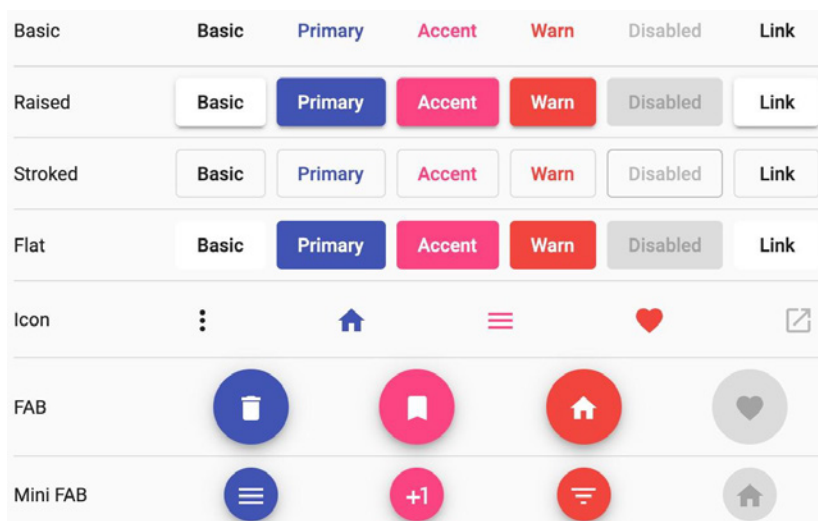
Ovo načelo upotrebljivosti naglašava konzistentnost u dizajnu korisničkog interfejsa kao faktor koji čini interfejs lakšim za učenje. Slično tome, naučeni interfejs se brže i sigurnije koristi. Značenje proizilazi iz konzistentnosti kao principa dizajna - identični termini, situacije i akcije moraju imati isto značenje u okviru interfejsa; različiti termini, situacije i akcije moraju imati različito značenje.

Konzistentnost ima dva nivoa - nivo prikaza i nivo ponašanja. Oba nivoa su podjednako važna i moraju biti realizovani dizajnom interfejsa. Konzistentnost na nivou prikaza podrazumeva konzistentan izgled različitih mesta aplikacije (strane ili pogledi u zavisnosti od načina realizacije). Ovde se misli na zajedničke elemente kao što su boje, oblici, terminologija veličina i raspored elemenata, nezavisno od specifičnog sadržaja strana interfejsa. Konzistentnost na nivou ponašanja znači da funkcije moraju biti identične u pogled ulaza korisnika, obrade (značenje i redosled koraka) i rezultata koje daju nezavisno od toga u kom delu aplikacije se koriste. Standardne aplikacije ispoljavaju konzistentan interni izgled i raspored funkcija.



Slika 106. Konzistentnost i standardi - *Spotify* aplikacija za slušanje muzike pruža doživljaj konzistentnog izgleda i ponašanja prilikom korišćenja različitih prikaza interfejsa.

Biblioteke i okviri za razvoj korisničkih interfejsa primenjuju standardne obrasce izgleda i ponašanja elemenata (kao što su dugmad, meniji, horizontalne i vertikalne trake za navigaciju i drugi). Time doprinose konzistentnosti u razvoju korisničkog interfejsa.



Slika 107. Konzistentnost i standardi - Praktični standardi izgleda i ponašanja elemenata korisničkog interfejsa koje propisuju okviri za razvoj doprinose konzistentnosti realizovanih aplikacija.

Izvor: <https://material.angular.io/components/button/examples>.

9.1.5. H5: Prevencija grešaka

Ovaj princip ističe razvoj korisničkog interfejsa koji uzima u obzir prevenciju grešaka pre nego one nastanu. Ovde se pre svega misli na logiku rada interfejsa koja je osmišljena i dizajnirana tako da eliminiše uslove i situacije koji mogu izazvati grešku, i traži potvrdu pre izvršenja kritičnih akcija.

Kao što je opisano u Poglavlju 3.3.1. najčešće greške u korisničkim interfejsima su greška sličnosti akcija (dve akcije izgledaju slično i korisnik greškom pokreće jednu umesto druge) i greška višeznačne akcije (jedna akcija ima više značenja na različitim mestima u aplikaciji). Ove greške su direktna posledica neodgovarajućeg dizajna elemenata interfejsa koji pokreću akcije. To su uglavnom dugmad, ikone i stavke menija. Ovaj problem može biti otklonjen u toku samog razvoja tako što se interfejs može povremeno analizirati sa aspekta dve vrste greške koje se često javljaju.

Pravila i smernice za realizaciju ovog principa su opisani u Poglavlju 3.3. Pored toga, pridržavanje ostalih heuristika upotrebljivosti vodi prevenciji grešaka u korišćenju interfejsa.

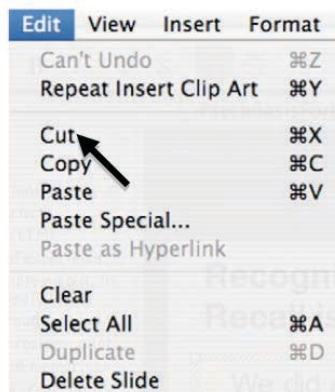
9.1.6. H6: Prepoznavanje pre pamćenja

Heuristika je formulisana na osnovu modela procesiranja čoveka (Poglavlje 4.1) i istoimenog principa principa lakoće učenja kao atributa upotrebljivosti

(Poglavlje 3.1.1). Mehanizam razmene informacija između kratkoročne i dugoročne memorije funkcioniše po principu asocijacija ili tragova informacija. Ukoliko su tragovi informacije prisutni u radnoj memoriji, informacija se brže i lakše dobavlja iz dugoročne memorije. Tako je, na primer, lakše odgovoriti na pitanje iz skupa ponuđenih odgovora (prepoznavanje) nego formulisanjem odgovora (pamćenje).

Interfejs je lakši za korišćenje ako se optimizuje mentalno naprezanje korisnika. Potrebno je učiniti objekte, akcije i opcije vidljivim i naznačenim na interfejsu. Element korisničkog interfejsa se lakše prepoznaje ukoliko postoji vizuelni trag o njemu. Sa jedne strane, ovo je slučaj sam po sebi sa obzirom da GKI koriste vizuelne elemente. Međutim, skup informacija interfejsa treba kreirati tako da je svaki element uočljiv u okviru celine. Ovo podrazumeva upotrebu prikladnih boja, oblika, veličina i naziva/simbola dugmadi, ikona, stavki menija i drugih kontrola interfejsa tako da ih korisnici lako prepoznaju (Slika 108). Na ovaj način se pre svega redukuju greške u pogledu izbora neodgovarajuće opcije interfejsa za dati cilj korisnika.

> copy doc1 doc2
> remove olddoc

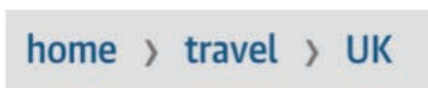


Slika 108. Prepoznavanje pre pamćenja - videti i izabrati (desno) je lakše nego pamtiti i kucati (levo).

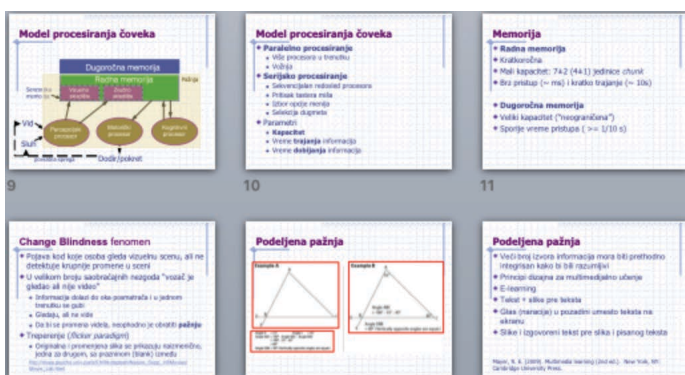
Poželjno je koristiti vizuelne simbole kao kombinaciju znakova, oblika i boja za predstavljanje funkcija ili sadržaja korisničkog interfejsa. Treba voditi računa da značenje simbola asocira na element u pozadini koji predstavlja (Slika 109).



(a)



(b)



(c)

Slika 109. Prepoznavanje pre pamćenja - pravilno dizajnirani vizuelni simboli akcija i sadržaja interfejsa pospešuju prepoznavanje (a i b). Pored toga simboli se mogu izvesti iz sadržaja kako bi korisnik imao bolju sliku o celini bez pretraživanja, minijature slike (eng. *thumbnail*) pod (c).

9.1.7. H7: Fleksibilnost i efikasnost korišćenja

Poznato je da svaki korisnički interfejs može imati tri kategorije korisnika - nove korisnike bez iskustva korišćenja interfejsa, povremene korisnike, i iskusne korisnike [79]. Načelo upotrebljivosti govori da sistem u pogledu fleksibilnosti i efikasnosti mora biti prilagođen korisnicima bez iskustva, i korisnicima sa iskustvom korišćenja. Ako govorimo o efikasnosti korišćenja, heuristika je podržana smernicama dizajna opisanim u Poglavlju 3.2.

Kao što je opisano u Poglavlju 4.4, ljudska percepcija je prirodno ustrojena da vidi strukturu koju formiramo prema Geštalt principima organizacije informacija. Principi blizine, sličnosti i simetrije se često koriste u dizajnu GKI. Oni pozitivno utiču na efikasnost korišćenja interfejsa.

Princip blizine se realizuje kao približavanje ili udaljavanje elemenata interfejsa radi bolje vizuelne organizacije sadržaja. Fizički bliske stvari se opažaju kao povezane, dok se fizičke udaljene stvari opažaju kao nepovezane. Može se reći da je logička povezanost među elementima interfejsa funkcija međusobne udaljenosti:

$$POVEZANOST = F(UDALJENOST),$$

gde su udaljenost i povezanost obrnuto srazmerne (manja udaljenost znači veću povezanost i obrnuto).

U korisničkom interfejsu treba približiti (grupisati) povezane elemente, i razdvojiti elemente koji nisu povezani. Kriterijumi povezanosti su određeni namenom interfejsa i vrstama akcija korisnika, odnosno elementima interfejsa koji realizuju akcije. Ako se radi o tekstualnim poljima za unos i pridruženim labelama, kriterijum su tipovi i značenje podataka koji se prosleđuju aplikaciji. Na primer, kontakt podaci i podaci o proizvodu koji se kupuje kod forme za onlajn kupovinu, ili grupe filtera informacija (vrsta proizvoda, vreme distribuiranja, cena, proizvođač, ili recenzije naručilaca) kod forme za pretragu (Slika 110).

3D

3D

Snaps | Art HD
Sci-Fi / Industrial by Manufactura K4

Heroic Fantasy Creatures Full Pack Volume 1
30 fantastic beasts in one pack

Sub-categories

Animations Characters Environments GUI Props Vegetation Vehicles

1-24 of 35918 results

Sort by Popularity View Results 24

3D

Refine by [clear filters](#)

☐ Hide Purchased Assets

All Categories

- ☒ 3D (35918)
- ☐ 2D (8327)
- ☐ Add-Ons (44)
- ☐ Audio (6248)
- ☐ Essentials (50)
- ☐ Templates (2944)
- ☐ Tools (8873)
- ☐ VFX (2549)

Pricing

\$ 0 - \$ 1500

☐ Free Assets (2605)

Unity Versions +

Publisher +

Ratings +

Platforms +

Release Date +

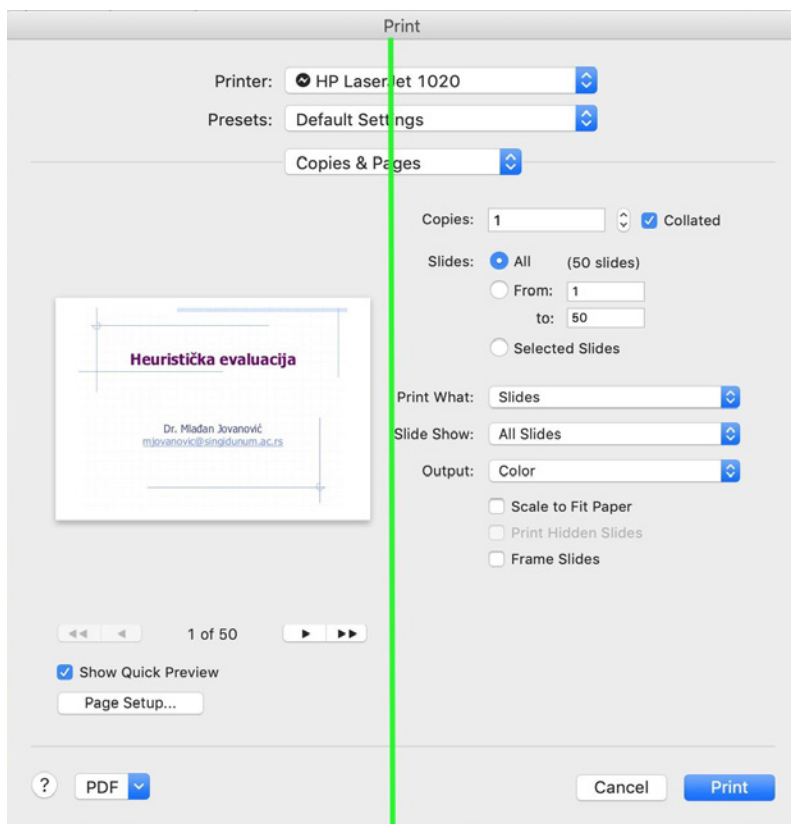
Slika 110. Fleksibilnost i efikasnost korišćenja - forma za pretragu u desnoj vertikalnoj traci interfejsa grupiše kriterijume pretrage po kategorijama podataka. Izvor: *Unity Assets Store*.

Ako govorimo o dugmadima i ikonama za pokretanje akcija interfejsa, tada je kriterijum logička povezanost ili sličnost akcija koje se pokreću datim elementima (Slika 111).



Slika 111. Fleksibilnost i efikasnost korišćenja - Dugmad u zaglavlju interfejsa su grupisana na osnovu funkcija koje obavljaju kao raspored interfejsa, dodavanje/uklanjanje particija, upravljanje particijom, i informacije o izabranoj particiji.

Simetričan raspored sadržaja doprinosi bržem vizuelnom skeniranju interfejsa, odnosno povećava efikasnost. Važno je napomenuti da simetrija interfejsa nije apsolutna već podrazumeva balans količine sadržaja (Slika 112).



Slika 112. Fleksibilnost i efikasnost korišćenja - simetrija kao balans sadržaja i ravnanje elemenata korisničkog interfejsa.

Ravnanje doprinosi povezivanju informacija interfejsa što dalje pozitivno utiče na efikasnost. Stvara vizuelne konekcije prikazanih elemenata čime olakšava skeniranje interfejsa. Ravnanje može biti horizontalno i vertikalno.

Kod horizontalnog ravnanja labele se mogu ravnanati po levoj ili desnoj strani. Poželjno je da budu približno iste dužine (ukoliko je izvodljivo). Dugmad i tekstualna polja treba poravnati po levoj strani.

Kod vertikalnog ravnanja labele i kontrole koje one opisuju moraju biti u istom redu i imati istu visinu osnovne linije teksta koji sadrže (eng. *text baseline*).

Pravilno korišćenje praznog prostora je važno za tekstualni sadržaj interfejsa [80]. Zbog toga što sadržaj GKI najpre vizuelno skeniramo a zatim analiziramo detalje (odnosno čitamo) [50], horizontalni prostor između reči mora omogućiti brzo i precizno prepoznavanje oblika reči. Suviše veliki prostor može ometati ritam čitanja, Slično tome, prostor između linija teksta mora omogućiti brzo i

precizno prepoznavanje oblika reči po vertikali. Suviše veliki prostor otežava vertikalno skeniranje teksta.

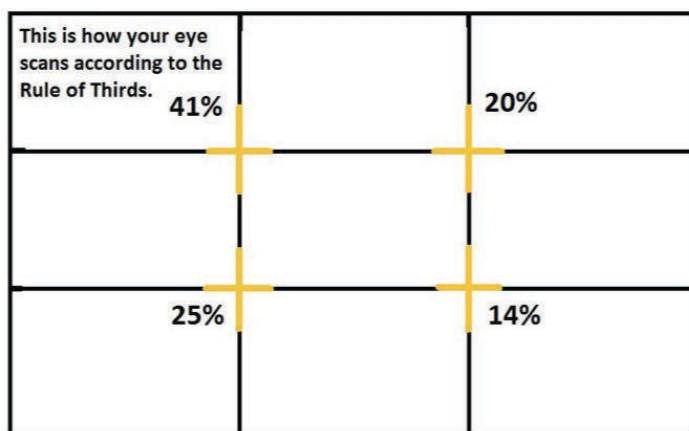
Uvek koristiti margine i ne stavljati tekst uz ivice prikaza zbog toga što će se teže opaziti.

Za tekst paragrafa je poželjno koristiti *Serif* fontove (na primer *Times* ili *Georgia*) zbog toga što oštrije forme za oblike reči pospešuju skeniranje i čitanje.

Za izolovane delova teksta (naslovi i zaglavlja) koristiti *Sans Serif* fontove (na primer *Verdana* ili *Arial*) zbog specifičnih oblika koji dodatno poboljšavaju kontrast u odnosu na tekst paragrafa (uz isticanje veličinom i pozicijom).

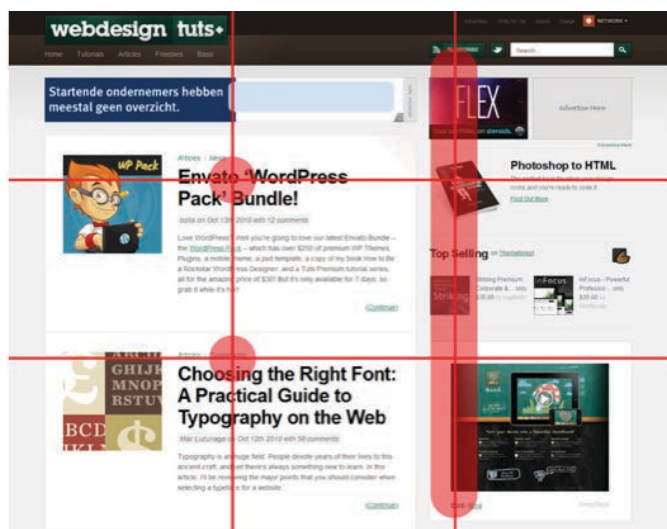
U pogledu rasporeda informacija u GKI, mogu se primenjivati dva pravila vizuelnog dizajna. Pravila su izvedena empirijski, eksperimentalnom analizom načina na koji korisnici skeniraju i čitaju GKI. Na ovaj način pravila mogu pomoći u raspoređivanju sadržaja korisničkog interfejsa kako bi relevantniji delovi bili istaknutiji.

Umetnik Džon Smit je 1797. godine upotrebio termin Pravilo Trećina (eng. “*Rule of Thirds*”) u svom delu Zapažanja o Ruralnim Predelima (eng. “*Remarks on Rural Scenery*”). Smit je naveo da se platno pri slikanju može posmatrati kao mreža kako bi se posmatračima pažnja privukla na specifična mesta slike. Pravilo ima široku primenu u vizuelnom dizajnu i dizajnu GKI. Ako se prikaz interfejsa ravnomerno podeli horizontalnim i vertikalnim osama na tri dela, tada se u preseccima osa nalaze fokusne tačke (eng. *focal points*) koje definišu vremena zadržavanja pogleda korisnika (Slika 113). Pravilo može pomoći u raspoređivanju elemenata korisničkog interfejsa.



Slika 113. Pravilo Trećina definiše fokusne tačke ekrana u čijim okolinama se mogu smestiti specifični sadržaji interfejsa u zavisnosti od njihove važnosti. Preuzeto iz [81].

Slika 114 analizira postojeći dizajn sa aspekta Pravila Trećina.



Slika 114. Pravilo Trećina se koristi u dizajnu korisničkog interfejsa za raspoređivanje elemenata sadržaja.

Izvor: <https://webdesign.tutsplus.com/articles/mathematics-and-web-design-a-close-relationship--webdesign-1053> (15.11.2020).

Sledeće pravilo je rezultat istraživanja praćenja pogleda N=263 korisnika prilikom čitanja Veb strana [82]. Rezultati pokazuju da obrazac pokreta čitanja u većini slučajeva poprima približan oblik slova F - vertikalna traka i horizontalne trake čija se dužina smanjuje vertikalno (Slika 115). Rezultati su dobijeni za interfejse kod kojih preovlađuje tekstualni sadržaj, međutim skorija istraživanja pokazuju da se obrazac javlja i kod mobilnih korisničkih interfejsa [80]. Važno je istaći da obrazac nije zajednički za sve korisnike. Drugim rečima, ne treba računati da će svi korisnici interfejsa koji razvijamo na taj način čitati sadržaj. Pored toga, kao takav ne mora predstavljati optimalno rešenje za različite vrste sadržaja interfejsa [82]. Sa druge strane, princip je posledica načina na koji čitamo bilo koji sadržaj i ima značajne implikacije na dizajn korisničkog interfejsa (naročito za tekst). Pre svega, početni paragrafi (obično prva dva) moraju sadržati najvažnije informacije. Većina korisnika će pročitati prvi paragraf i skenirati drugi paragraf. U nastavku teksta, podnaslove, paragrafe i liste treba početi rečima sa istaknutijim značenjem koje prenose poruku ili opisuju čitav paragraf. Razlog tome je što će korisnici opaziti ove reči vertikalnim skeniranjem na dole po levoj strani interfejsa (glavna drška slova F). Treća i naredne reči u liniji će biti pročitane značajnije ređe od prve dve reči.



Slika 115. F obrazac skeniranja sadržaja GKI opisuje kombinacije vertikalnih i horizontalnih pokreta pogleda korisnika prilikom čitanja sadržaja interfejsa.

Izvor: <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content-discovered/> (15.11.2020).

9.1.8. H8: Estetski i minimalistički dizajn

Ovaj princip upotrebljivosti govori da interfejs ne treba sa sadržaj informaciju koja nije relevantna ili se retko koristi. U isto vreme, interfejs mora biti estetski dopadljiv korisnicima. Drugim rečima dizajn interfejsa mora uspostaviti balans između tendencije dodavanja sadržaja (u cilju veće privlačnosti interfejsa) i redukovanja sadržaja (u cilju zadržavanja relevantnih informacija).

Ova heuristika upotrebljivosti prvenstveno poboljšava učenje korišćenja interfejsa, a zatim redukuje greške i povećava efikasnost.

Prilikom dizajna interfejsa svaka nova informacija privlači dodatnu pažnju korisnika i može skrenuti fokus sa postojećih, bitnih informacija. Jednostavnost (eng. *simplicity*) kao princip vizuelnog dizajna nalaže dodavanje što je moguće manje elemenata tako da čine skladan i kompletan izgled i funkcionalnost interfejsa u skladu sa namenom. U suštini može se govoriti o procesu eliminacije, a ne dodavanja. Ovo može biti u suprotnosti sa tendencijom dizajnera da dodaju elemente (“za svaki slučaj” ili “samo za slučaj da ...” kao tipični obrasci razmišljanja). Zbog toga je gore pomenuti balans ključan za upotrebljiv krajnji rezultat.

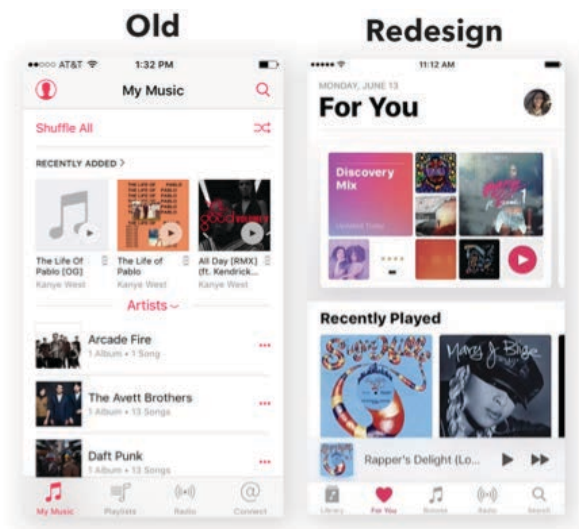
Jednostavnost je složen koncept koji se implementira korišćenjem specifičnih tehnika u dizajnu korisničkog interfejsa. Tri često korišćene tehnike su [29]:

- Redukcija (eng. *reduction*),
- Regularnost (eng. *regularity*),
- Višestruke uloge (eng. *multiple roles*).

Redukcija predstavlja uklanjanje elemenata koji nisu neophodni za funkcionisanje interfejsa. Često se vrši naknadno, nakon što je određena strana interfejsa realizovana. Zahteva kritičku analizu elemenata interfejsa. U tom pogledu, opšte smernice za analizu sadržaja strane interfejsa mogu biti:

1. Definirati cilj dizajna strane interfejsa kao poruku koju prenosi korisniku.
2. Kritički ispitati da li svaki element interfejsa služi datom cilju.
3. Ukloniti sve elemente koji ne služe cilju.
 - a. U slučaju da niste sigurni, uklonite element i vidite da li dizajn prenosi istu poruku iz tačke 1.

Slika 116. daje primer redukcije elemenata identične strane korisničkog interfejsa različitih verzija aplikacije *Apple Music*.

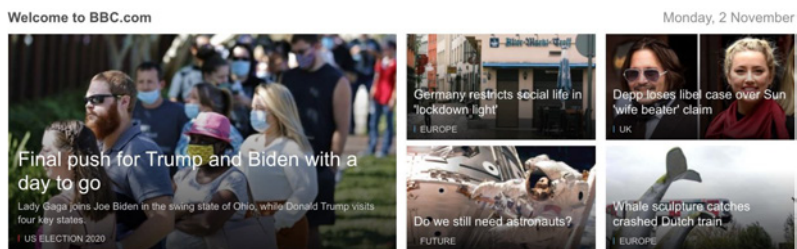


Slika 116. Estetski i minimalistički dizajn - Jednostavnost kao redukovanje elemenata korisničkog interfejsa aplikacije za slušanje muzike.

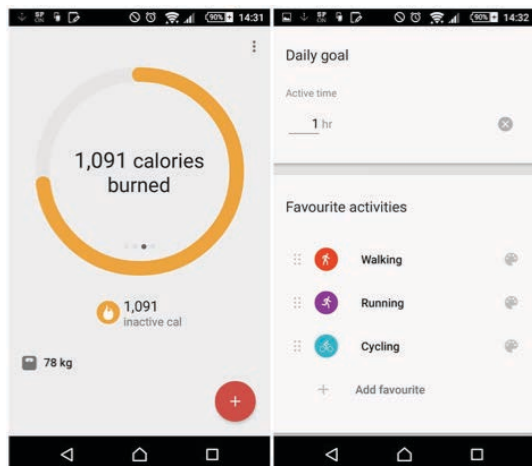
Izvor: <https://medium.com/amazing-stuff/complexion-reduction-a-new-trend-in-mobile-design-cef033a0b978> (15.11.2020).

Regularnost (ili ponavljanje) podrazumeva ponavljanje specifičnih elemenata interfejsa. Obično se definiše šablon prikaza elemenata i ponavlja na jednom ili više mesta (strana) interfejsa (Slika 117). Šablon može biti složen i predstavljati kombinaciju vizuelnih elemenata interfejsa poput fontova (teksta), boja, linija, veličina, oblika i orijentacija. Ako je dizajn regularan, sa jedne strane daje utisak jednostavnosti, dok u isto vreme omogućava isticanje važnijih delova (Slika 117a) interfejsa.

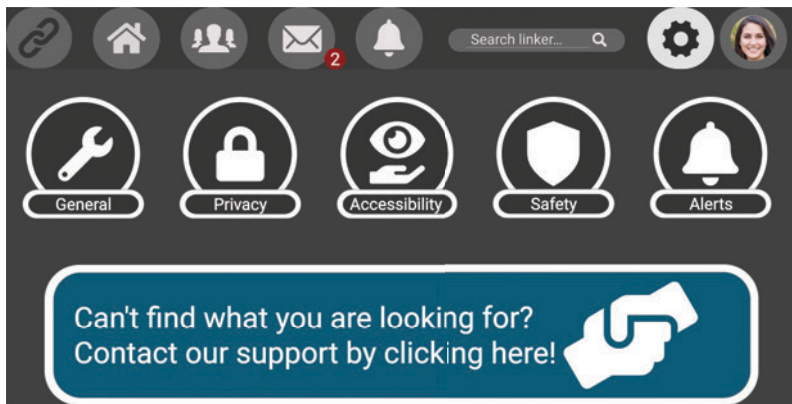
Dok kontrast ističe razlike, regularnost daje utisak pripadanja celini kao uniformnost, konzistentnost i koheziju. Tehniku treba koristiti pažljivo zbog toga što previše ponavljanja može proizvesti utisak monotonog, neinteresantnog dizajna.



(a)

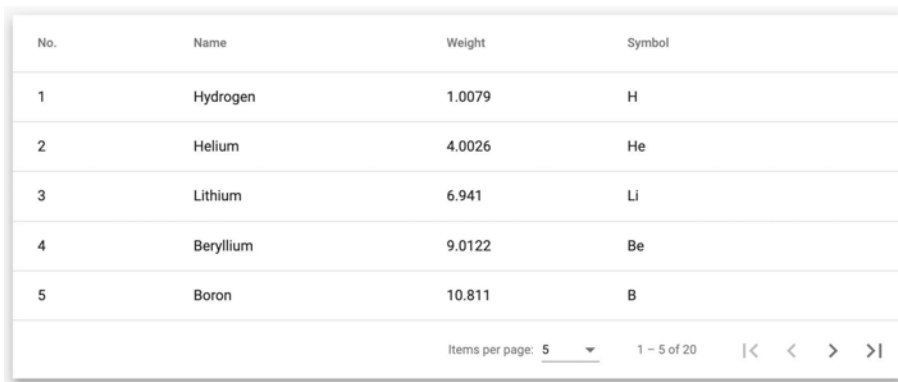


(b)



Slika 117. Regularnost u estetskom i minimalističkom dizajnu - Ponavljajući elementi se mogu istaći pozicijom i veličinom iz skupa (a). Ponavljajući šablon koji kombinuje elemente interfejsa opisuje omiljene aktivnosti korisnika u donjem delu interfejsa na desnoj strani (b). Ponavljajuće forme ikona sa labelama u središnjem delu (c). Preuzeto iz [83].

Tehnika *višestruke uloge* znači da element korisničkog interfejsa može imati veći broj funkcija, odnosno uloga. Na primer, element putokaz (eng. *breadcrumb*) (Slika 109b) ima dve funkcije - orijentacija (govori korisniku gde se trenutno nalazi u aplikaciji) i navigacija (omogućava korisniku da se kreće kroz aplikaciju). Slično tome, tabela kao kontrola GKI sadrži podnožje (eng. *footer*) kao element koji pokazuje veličinu i tekuću stranu tabele, omogućava promenu veličine strane tabele, i kretanje kroz strane tabele (Slika 118).



No.	Name	Weight	Symbol
1	Hydrogen	1.0079	H
2	Helium	4.0026	He
3	Lithium	6.941	Li
4	Beryllium	9.0122	Be
5	Boron	10.811	B

Items per page: 5 1 - 5 of 20 < > < >

Slika 118. Estetski i minimalistički dizajn - podnožje tabele predstavlja element koji ima veći broj funkcija, odnosno uloga.

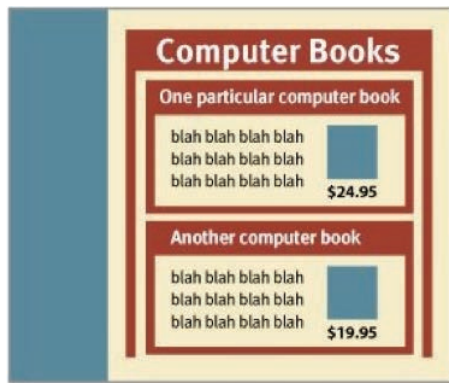
Izvor: <https://material.angular.io/components/table/examples> (15.11.2020).

Kontrast je važan element estetskog i minimalističkog doživljaja korisničkog interfejsa. Informacije interfejsa moraju biti pravilno struktuirane tako da pridobiju odgovarajuću pažnju korisnika. Elementi koji treba da privuku pažnju moraju biti istaknuti u odnosu na druge elemente. U dizajnu GKI, kontrast podrazumeva kreiranje vizuelnih hijerarhija elemenata. Hijerarhije uspostavljaju razlike u vizuelnim varijablama (eng. *visual variables*) koje korisnik opaža. Razlike vizuelno ističu elemente za koje smatramo da su važniji. Primeri vizuelnih varijabli su boja (eng. *value*), nijansa (eng. *hue*), oblik (eng. *shape*), pozicija (eng. *position*), orijentacija (eng. *orientation*) i veličina (eng. *size*) [84].

Slika #a daje šemu hijerarhije tekstualnog sadržaja koja koristi razlike dve vizuelne variable - pozicija i veličina. Koreni nivo neposredno privlači pažnju korisnika (na primer naslov vesti ili članka) i daje osnovnu informaciju. Srednji nivo pruža dodatne informacije kako bi olakšao skeniranje sadržaja. Ovo mogu biti meta-podaci vesti ili članka poput datuma, autora i ključnih reči. Treći nivo daje detalje sadržaja (tekst vesti) koji će korisnik čitati nakon što nađe vest ili članak od interesa.



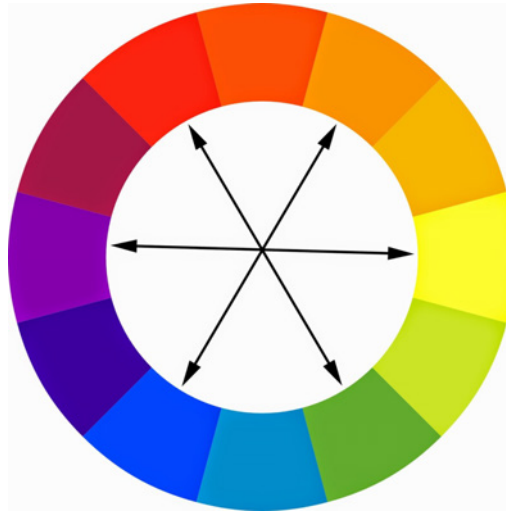
(a)



(b)

Slika 119. Estetski i minimalistički dizajn - (a) kontrast pomoću pozicije i veličine, i (b) kontrast pomoću pozicije, veličine, boje i oblika. Preuzeto iz [50].

Boja je često korišćena vizuelna varijabla za postizanje kontrasta. Na primer, boja pozadine i teksta dugmeta mora imati dobar kontrast. Točak boja (eng. *color wheel*) daje pogodne kombinacije boja za postizanje kontrasta u vizuelnom dizajnu [85]. Neke kombinacije boja nisu prikladne za korisnike sa posebnim potrebama. Treba kombinovati boje koje nisu susedne, odnosno koje su međusobno dijametralno raspoređene (Slika 120a).



(a)



(b)

Slika 120. Točak boja za realizaciju vizuelnog kontrasta - (a) kombinovanje osnovnih boja, i (b) kombinovanje nijansi osnovnih boja.

Izvor: <https://www.canva.com/colors/color-wheel/> (15.11.2020).

9.1.9. H9: Prepoznavanje i oporavak od grešaka

Dok se heuristika H5 odnosi na sprečavanje grešaka pre nego one nastanu, ova heuristika ističe važnost informisanja korisnika i nastavka korišćenja sistema nakon pojave greške. U tom pogledu pravila za prepoznavanje (kao informisanje korisnika) i oporavak od greške su identična kao u Poglavljima 3.3.3 i 3.3.4.

9.1.10. H10: Pomoć i dokumentacija

Iako je poželjno da se korisnički interfejs može koristiti bez dokumentacije, ona može biti potrebna. Dokumentacija mora biti laka za pretragu, povezana sa zadatkom korisnika, konkretna, i ne previše obimna. Smernice za realizaciju pomoći u tekstualnom obliku su opisane u Poglavlju 3.1. Pored toga, video instrukcije mogu biti korisne ukoliko su vidljive i pristupačne na interfejsu, što je moguće kraće, i ako prikazuju samo informacije specifične i relevantne za izvršenje zadatka.

9.2. Elementi heurističke evaluacije

Proces heurističke evaluacije se sastoji iz tri faze:

- Informisanje korisnika,
- Testiranje korisničkog interfejsa u odnosu na heuristike upotrebljivosti,
- Analiza rezultata.

Na početku se korisnici upoznaju sa interfejsom i zadacima. Zadaci i heuristike se obično daju u papirnoj formi. Opšti zadatak korisnika je da otkriju probleme upotrebljivosti i da ih konkretno opišu.

Nakon toga, korisnici obavljaju zadatke na interfejsu. Važno je napomenuti da zadaci služe za orijentaciju, odnosno da pored toga ispitivači imaju slobodu u korišćenju interfejsa i testiranja i drugih funkcija. Ukoliko se angažuju eksperti upotrebljivosti, oni mogu i samostalno istraživati interfejs i pronalaziti probleme. Ovi korisnici mogu ispitivati interfejs i u većem broju iteracija. U praksi su to obične dve iteracije. Prva iteracija ima za cilj upoznavanje sa interfejsom. Samim tim, u drugom prolasku se mogu bolje pronaći problemi upotrebljivosti interfejsa. Kada se angažuju korisnici koji nisu eksperti, njima je poželjno dati zadatke kako bi testiranje bilo efektivnije i efikasnije.

U slučaju testiranja funkcionalnog prototipa, poželjno je definisati zadatke za korisnike kako bi ispitivanje bilo fokusirano na relevantne elemente interfejsa. Pravila za formulisanje zadataka su identična kao kod tradicionalnog testiranja

upotrebljivosti (Poglavlje 8.3.1). U slučaju testiranja ranog prototipa, korisnici mogu slobodno istraživati interfejs.

Analiza rezultata podrazumeva konačno definisanje problema upotrebljivosti i njihovih prioriteta. Nivoi ozbiljnosti problema se definišu slično kao kod testiranja upotrebljivosti sa krajnjim korisnicima (Poglavlje 8.3.1), u zavisnosti od učestalosti i uticaja problema na korišćenje interfejsa. Rezultat se može opisati u izveštaju koji sadrži sledeće elemente:

- Opis problema (konkretna i sažeta), sa kopijama izgleda ekrana kao opcijom.
- Heuristika koja je narušena za svaki problem (jedna ili više).
- Ozbiljnost problema, gde nivo određuju ispitivači u zavisnosti od identifikovanih problema.
- Preporuke za otklanjanje problema.

Heuristička evaluacija može biti efikasan i efektivan metod za testiranje upotrebljivosti GKI. Sa druge strane, ne može u potpunosti zameniti tradicionalno testiranje upotrebljivosti opisano u Poglavlju 8. Jedan od problema može biti iskustvo, odnosno znanja i veštine korisnika koji mogu uticati na predrasude u prikupljenim podacima. Na primer, eksperti se mogu više fokusirati na elemente interfejsa koji su relevantni sa aspekta njihovog ličnog iskustva. Ovde se mogu javiti takozvani nepostojeći problemi (eng. *false alarms*) i neotkriveni problemi (eng. *missed problems*). Ovi negativni efekti se mogu ublažiti angažovanjem većeg broja korisnika i komplementarnim testiranjem (kombinacija sa tradicionalnim testiranjem upotrebljivosti).

10. Vrste i elementi GKI

Rani računarski sistemi bili su usko specijalizovani, odnosno koristili su se za vrlo specifičnu namenu - na primer, razbijanje šifrata određenog algoritma ili specifičnu matematičku funkciju za potrebe nuklearnih fizičara. Takve računare koristili su visoko obrazovani korisnici, najčešće nosioci doktorata iz matematike, fizike ili tehničkih nauka, koji su do detalja razumevali na koji način računar koji koriste rešava određeni problem. Šta više, sami korisnici su često bili i tvorci računara koje koriste, tako da nije bilo potrebe za obukama i uputstvima za korišćenje, ili intuitivnim interfejsima.

Okončanjem Drugog svetskog rata potreba za korišćenjem računara u vojne svrhe znatno je opala. Računari, sada korisna pomagala u naučnim istraživanjima, preselili su se na univerzitete. Međutim, prepoznat je ogroman potencijal za njihovu primenu i u poslovnim okruženjima.

Jedan od osnovnih problema za primenu računara u poslovnim okruženjima bio je nedostatak tehničkog znanja i obrazovanja ciljnih korisnika. Tipičan ciljni korisnik bio je, na primer, računovođa u velikoj kompaniji, koji nije imao nikakva znanja o algoritamskom rešavanju problema, binarnom brojnem sistemu, ili mehaničkim i električnim komponentama. Iz tog razloga bilo je neophodno apstrahovati komunikaciju korisnika sa računarom, odnosno pojednostaviti način na koji korisnik zadaje naredbe računaru i na koji od njega dobija rezultate. U tom pogledu, razvoj računara je nastavljen u vidu podizanja nivoa apstrakcije i intuitivnosti u načinu komunikacije sa korisnikom i razvoja hardverskih komponenta koje podržavaju takvu interakciju.

Kao što je za definisanje novog, apstraktnog objekta potrebno napraviti vezu, odnosno definisati odnos između njega i postojećeg znanja, tako su i računarski korisnički interfejsi pratili principe skeumorfizma (eng. *skeuomorphism*). Termin ima slično značenje kao metafore i priuštivost (Poglavlja 1.4.4 i 1.4.6) i označava izvedeni objekat koji zadržava karakteristike originala [86]. Pošto je osnovni objekat za interakciju u poslovnim okruženjima bila pisaća mašina, za inicijalnu komunikaciju sa računarskim sistemom razvijeni su uređaji koji su, sa aspekta korisnika, izgledali njoj gotovo identično. Za zadavanje naredbi računaru razvijene su tastature (taster *Enter* dugo je obeležavan kao *Return*, po uzoru na deo pisaće mašine namenjen za prelazak u novi red), a rezultati izvršavanja naredbi štampani su na papiru.

Nasleđe vezano za zasnovanost na mehaničkim pisaćim mašinama ima svoju cenu. Na primer, popularni raspored slova *QWERTY* razvijen je da bi se smanjilo zaglavljivanje mehaničkih elemenata pisaćih mašina. Sa obzirom da kod elektronskih tastatura takva vrsta zaglavljivanja nije moguća, alternativni

rasporedi slova na tastaturi (na primer, Dvorak ili Kolmak) mogu povećati brzinu kucanja i ergonomiju. Međutim, troškovi prelaska na te rasporede, u vidu zamene ili izmene postojećih tastatura, kao i vremena potrebnog da se korisnici priviknu na njih, preveliki su da bi takva migracija bila izvodljiva.

Na dalje, osnovni interfejs bio je unapređivan u različitim pravcima. Štampači i papir zamenjeni su TV ekranima. Umesto interpretera linijskih naredbi razvijene su sofisticiranije softverske komponente korisničkog interfejsa, koje su omogućile upotrebu primitivnih grafičkih komponenata i miša. Moćniji centralni procesori, uvođenje specijalizovanih grafičkih procesora i ekrani sa većom rezolucijom i osetljivi na dodir omogućili su razvoj naprednih grafičkih interfejsa prema korisnicima.

Zvuk, kao jedan od osnovnih nosilaca komunikacije među ljudima, jedan je od načina za interakciju i sa računarima. Inicijalno, zvuk je uveden kao komponenta multimedijalnih interfejsa, prvenstveno u smeru od računara ka korisniku. Kasnije, razvojem dovoljno moćnih procesora i algoritama za prepoznavanje govora, omogućeno je korišćenje zvuka i za glasovno zadavanje naredbi računarima. Ovakav tip interakcije nije posebno zaživeo u poslovnoj upotrebi računara, s obzirom da nije dovoljno efikasan i precizan, zamorniji je za korisnika, ranjiviji je u pogledu bezbednosti i privatnosti i nije pogodan za korišćenje u bučnim okruženjima. Međutim, ovaj tip interakcije je izuzetno koristan u situacijama kada su ekstremiteti i pogled korisnika angažovani u drugim aktivnostima (na primer, vožnja automobila).

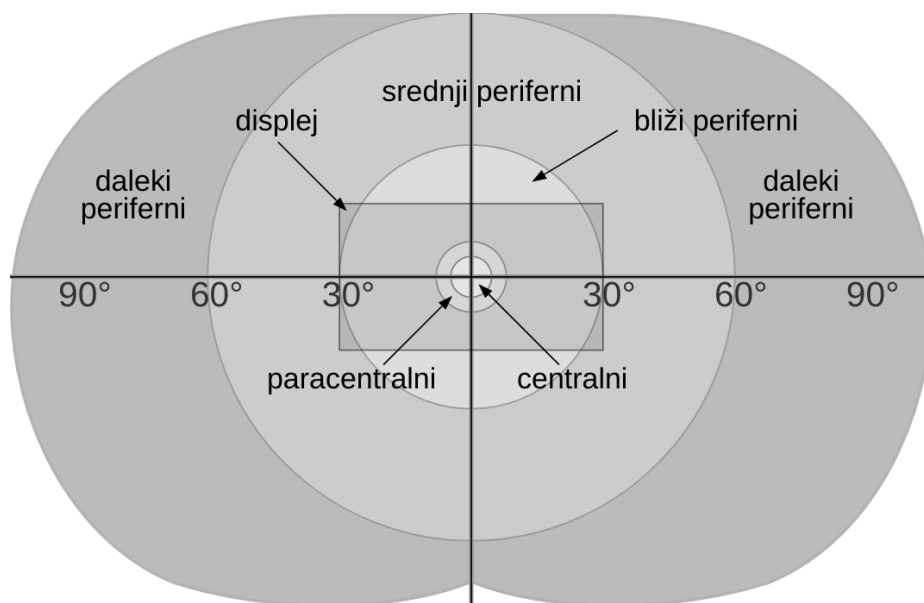
Vizuelna komunikacija prvenstveno je korišćena i koristi se u smeru od računara ka korisniku. Međutim, kao i u slučaju zvuka, razvoj procesora i algoritama doveo je do mogućnosti vizuelnog zadavanja naredbi računarima, korišćenjem kamera. Ovakav vid komunikacije često se koristi za dobijanje dopunskih informacija o korisniku (na primer prepoznavanje umora kod vožnje automobila ili emotivnog stanja korisnika). Sa druge strane, za korisnike koji imaju smanjenu mogućnost govora i korišćenja ekstremiteta, razvijeni su posebni sistemi za interakciju zasnovani na praćenju kretanja očnih jabučica i treptaja. Konačno, ovaj tip komunikacije omogućio je i biometrijsku autentifikaciju zasnovanu na prepoznavanju lica korisnika.

10.1. Grafički korisnički interfejs

U ovom delu opisujemo principe i arhitekturu GKI sa aspekta krajnjeg korisnika.

10.1.1. Čovekovo vidno polje i grafički prikaz

Vidno polje čoveka predstavlja oblast u kojoj čulo vida može da registruje prisustvo stimulusa. Međutim, oština vida, odnosno količina uočenih detalja, nije ista u svim delovima vidnog polja. Najveću oštrinu nudi centralni deo vidnog polja (u kojem oko može da razlikuje detalje nivoa jednog minuta, odnosno šezdesetog dela stepena), i ona opada ka periferiji. Ova karakteristika se ponekad koristi kod renderovanja (eng. *rendering*) scena 3D igara, gde se centralni deo prikaza renderuje u visokoj rezoluciji, a rezolucija opada prema perifernim delovima kako bi se uštedelo vreme grafičkih procesora. Takođe, u dizajniranju korisničkih aplikacija centralni deo ekrana se koristi za intenzivnu interakciju, dok se ređe korišćeni elementi interfejsa (na primer sat, obaveštenja, i slično) izmeštaju u periferne delove.



Slika 121. Struktura i oblasti čovekovog vidnog polja sa predlogom optimalne pozicije i veličine displeja.

Glavni parametri koje treba uzeti u obzir pri postavljanju optimalnog korisničkog interfejsa jesu dužina dijagonale displeja i odnos širine i visine, udaljenost od korisnika i oština prikaza. Osobina vidnog polja da ima veću širinu od visine uzeta je u obzir kod dizajniranja displeja, tako da većina ekrana danas prati odnos 16:9 (ili 21:9 kod ultra-širokih ekrana) između širine i visine. Ukoliko dijagonala prikaza nije srazmerna udaljenosti od korisnika, periferni delovi ekrana će pripadati perifernom vidnom polju i neće biti upotrebljivi ili će sam prikaz biti premali/predalek da bi se izdvojili potrebni detalji.

Oštrina, odnosno rezolucija prikaza prvenstveno zavisi od udaljenosti na kojoj se on nalazi u odnosu na korisnika. Potreban broj tačaka po inču se, za oštru percepciju prikazanog sadržaja, smanjuje u skladu sa udaljenošću. Tako, na primer, mobilni telefoni mogu imati i 600 DPI (eng. *dots per inch*) na ekranu dijagonale 6 inča, dok monitori sa 10 puta dužom dijagonalom mogu imati i srazmerno manji broj tačaka po inču, za istu percepciju oštine.

Ako uzmemo u obzir sve navedene parametre, možemo izračunati optimalnu konfiguraciju grafičkog prikaza. Na primer, za monitor dijagonale od 27 inča (odnos ivica 16:9, odnosno 598x336 mm) i 4K rezolucije (3840x2160 piksela, odnosno 163 tačke po inču ili oko 64,2 tačke po centimetru) optimalna udaljenost korisnika (za zauzimanje prostora bližeg perifernog vida) iznosi 519 mm. U slučaju da se rezolucija smanji na upola nižu vrednost, 1920x1080 piksela, udaljenost korisnika treba dvostruko povećati da bi se zadržao prethodni nivo (percepcije) oštine slike.

10.1.2. Sistemska arhitektura grafičkog korisničkog interfejsa

U ovom odeljku opisujemo najvažnije elemente arhitekture GKI.

10.1.2.1. Rezolucija i sinhronizacija prikaza

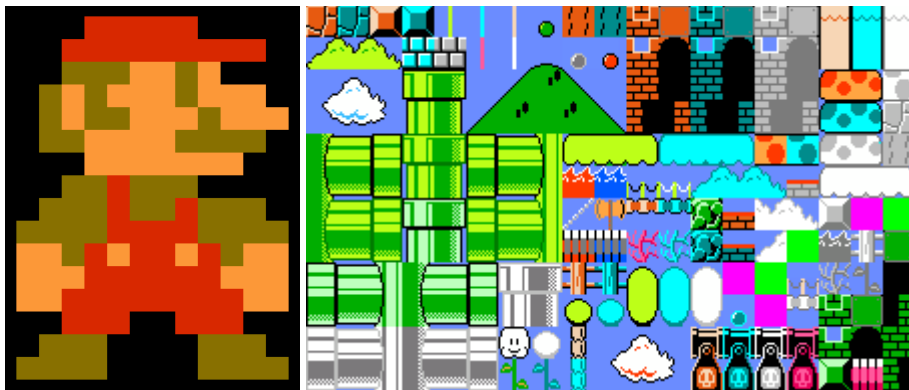
Prikaz na displeju predstavlja grafičku reprezentaciju - projekciju - aktuelnog stanja vizuelnih elemenata GKI. Ovi elementi su definisani kao specifične strukture podataka koje se nalaze u radnoj memoriji računara ili grafičkog adaptera. Neke od ovih struktura podataka sadrže upravo matrice piksela koje će direktno biti prikazane na displeju. Međutim, neke strukture sadrže matrice drugačijih rezolucija ili apstraktne (na primer vektorske) objekte, koje je pre prikazivanja potrebno prilagoditi mogućnostima samog displeja. Moguće su i promene samih displeja (na primer priključivanje novog monitora ili projektor, ili rotacija ekrana), koje takođe treba podržati.

Zatim, stanje objekata koje treba prikazati, odnosno izgleda GKI menja se tokom vremena. Ove promene mogu biti minimalne (pokretanje kursora miša ili pulsiranje pokazivača pozicije u programu za uređivanje teksta) i retke (kod čitanja teksta), ali i česte i obimne (kod igranja 3D video igara). Dakle, osim prostornog prilagođavanja prikaza, zadatak podsistema za GKI je i da vremenski sinhronizuje prikaz grafičkog sadržaja.

10.1.2.1.1. Prostorna rezolucija i sinhronizacija

Rani računarski sistemi imali su veoma ograničenu procesorsku moć i memorijski kapacitet, kao i ograničene mogućnosti prikazivanja grafičkih sadržaja. Iz tog razloga, korišćena je isključivo rasterska grafika, na diskretnim displejima, u niskoj rezoluciji i sa ograničenim brojem boja. Programiranje

korisničkog interfejsa podrazumevalo je zadavanje instrukcije grafičkom adapteru u kojoj boji treba prikazati piksel na određenim koordinatama. Sami displeji najčešće su bili standardizovani, kako bi se interfejs programa pravilno prikazivao u različitim sistemima.



Slika 122. Video igra *Super Mario Bros* iz 1985. godine kreirana je u rezoluciji 256x224 piksela i koristi 56 boja (standardno za *Nintendo Entertainment System* konzolu). Ukupna veličina igre je 31 KB.

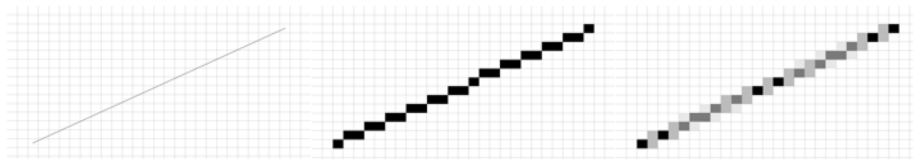
Počev od 80-ih godina 20. veka krenuo je intenzivan razvoj GKI. Broj i gustina piksela značajno su porasli, dok je broj boja dostigao 16 miliona, sa mogućnošću prozirnosti u slučaju preklapanja slojeva prikaza. Dodatno, procesorska moć i memorijski kapaciteti računara i grafičkih adaptera su konstantno unapređivani, što je otvorilo mogućnost za ozbiljniju primenu računara sa aspekta grafike, kao i za razvoj naprednijih GKI.

Međutim, nove mogućnosti dovele su i do pojave širokog spektra dimenzija i rezolucija ekrana. Na primer, danas je moguće istu aplikaciju koristiti na *Apple Watch* uređaju, čiji displej ima rezoluciju od 272x340 piksela, kao i na desktop računaru sa 4K monitorom (3840x2160 piksela). Ili, prenosivi računar *ThinkPad T14* ima ekran dijagonale 14" u različitim opcijama ima mogućnost maksimalne rezolucije od HD (eng. *high-definition*) (1366x768) do UHD (eng. *ultra-high-definition*) (3840x2160). Rasterska ikona prilagođena nižoj rezoluciji bila bi premala (gotovo tri puta prostorno manja) na ekranima više rezolucije, i obrnuto. Jasno je da definisanje korisničkog interfejsa i njegovih elemenata na nivou piksela više nije predstavljalo optimalno rešenje i bilo je potrebno pronaći nove principe za realizaciju GKI.

Prvi korak u podržavanju displeja različitih dimenzija i rezolucija bio je uvođenje vektorske grafike. Za razliku od rasterske grafike, elementi vektorske grafike se definišu kao vektori (oblici definisani matematičkim formulama), sa

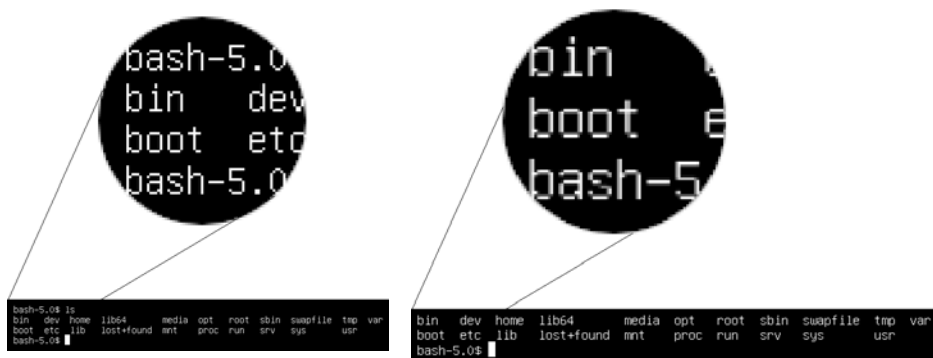
relativno zadatim dimenzijama. Prilikom iscrtavanja takvih elemenata na diskretnim displejima, oni se prevode u rastersku grafiku, odnosno izračunavaju se vrednosti diskretnih piksela u području dodeljenom za prikaz elementa. Sa vizuelnog aspekta rezultati ovog prevođenja obično su veoma dobri u slučajevima kada se vektorski elementi iscrtavaju na većoj površini, odnosno na površini sa velikim brojem piksela. Nasuprot tome, kada se vektorski element iscrtava na površini male rezolucije u pikselima, kao rezultat se može javiti “mutna” slika, odnosno slika na kojoj se ne vide potrebni detalji. Iz tog razloga se ponekad koriste kombinovani formati grafičkih elemenata, odnosno rasterska grafika za niže rezolucije, a vektorska za više.

Jedan od osnovnih problema kod diskretizacije, odnosno prevođenja vektorske u rastersku grafiku, jeste alijasing (eng. *aliasing*). Problem alijasinga dovodi to toga da susedni pikseli imaju iste diskretne vrednosti, iako originalni objekat ili vektorski zapis nemaju takvu karakteristiku. Za rešavanje ovog problema koriste se anti-alijasing funkcije koje, posebno na većim objektima, mogu dati dobre rezultate.



Slika 123. Iscrtavanje linije na diskretnom displeju: vektorska instrukcija (levo), sa alijasingom (centar) i sa anti-alijasingom (desno).

Jedan od prvih tipova grafičkih elemenata koji je preveden u vektorski oblik jesu fontovi. Rani računarski fontovi bili su definisani kao mape bitova, odnosno prilagođeni prikazu simbola na određenom polju piksela (na primer 8x16 piksela). Takav pristup je u potpunosti imao smisla, s obzirom da su displeji uglavnom imali standardnu rezoluciju od 640x480 piksela, odnosno 80x30 simbola. Međutim, promena veličine ovih fontova za potrebe ekrana veće rezolucije ili štampe, nije davala zadovoljavajući kvalitet po pitanju oštine.



Slika 124. Oštar prikaz rasterskog GNU unifont-a u predviđenoj veličini: 8x16 piksela (levo). Promena veličine fonta decimalnim množiocem smanjuje oštrinu i uvodi tonove sive boje (desno).

Nasuprot rasterskim fontovima, vektorski fontovi se mogu skalirati, odnosno mogu se štampati ili projektovati u različitim veličinama na displeje različitih rezolucija. U slučaju dovoljno visoke rezolucije displeja - takve da ljudsko oko nije u stanju da izdvoji pojedinačne piksele - simboli fontova mogu se prikazivati i sa alijasingom. U protivnom, potrebno je primeniti neku od tehnika za anti-alijasing.

Contrast

Contrast

Slika 125. Prikaz teksta vektorskim fontom u visokoj rezoluciji bez anti-alijasinga (gore) i sa anti-alijasingom (dole).

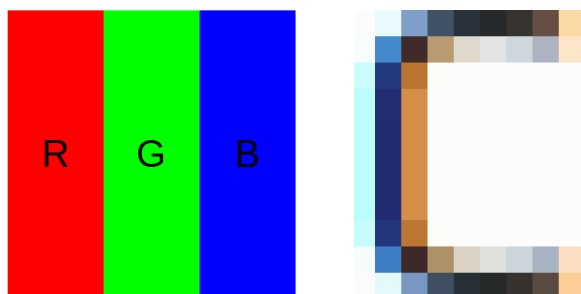
Na sledećoj slici prikazana je projekcija vektorskog fonta na displej niske rezolucije, odnosno vektorski definisani simboli su diskretizovani na mali broj piksela. Prilikom diskretizacije određeni pikseli - oni kroz koje prolaze konture vektorski definisanih elemenata - iscrtavaju se u zadatoj boji. Ukoliko se ne primenjuje neka od tehnika anti-alijasinga, pikseli mogu biti samo u zadatoj boji ili u boji pozadine, odnosno ivice simbola će biti “nazubljene” (gornji deo

slike). Međutim, primenom anti-alijasinga izračunavaju se tonovi boje, između boje pozadine i boje u kojoj se ispisuje simbol. Intenzitet boje se određuje na osnovu toga koliki deo piksela zahvata vektorski definisani simbol, odnosno koliko blizu njegovog centra prolazi vektorski definisana linija (donji deo slike).

Contrast
Contrast

Slika 126. Prikaz teksta vektorskim fontom u niskoj rezoluciji bez anti-alijasinga (gore) i sa anti-alijasingom (dole).

Alternativni način za podizanje rezolucije displeja, odnosno za iscrtavanje oštrije projekcije vektorskih simbola, jeste renderovanje ispod nivoa piksela (eng. *subpixel rendering*). Mogućnost za primenu ove metode čini fizička osnova prikazivanja piksela savremenih displeja. Na primer, pikseli LCD (eng. *liquid-crystal display*) se sastoje od tri horizontalno postavljene komponente, crvene, zelene i plave. U skladu sa tim, zadavanjem crvene boje određenom pikselu uključuje se njegova leva komponenta, a zadavanjem plave boje desna. Naravno, glavni nedostatak ovog pristupa je to što se u cilju podizanja rezolucije žrtvuje ispravnost prikazivanja boja.



Slika #. Jedan piksel u LCD sastoji se od tri vertikalno postavljene komponente, za emitovanje crvene, zelene i plave boje (levo). Izmenom boje piksela može se dobiti veća rezolucija prikaza, putem aktiviranja vertikalnih trećina pojedinačnih piksela (desno).

Još jedna popularna komponenta GKI koja je migrirala u vektorski oblik u savremenim korisničkim interfejsima jesu ikone. Setovi ikona ranije su obično

dizajnirani da podrže najčešće korišćene dimenzije, npr. 16x16, 24x24, 32x32 i 48x48 piksela. Međutim, ikone ovih dimenzija su ispod granice upotrebljivosti na savremenim displejima visokih rezolucija. Sa druge strane, njihovim skaliranjem, odnosno uvećavanjem, gubi se oštrina i jasnoća prikaza. Iz tog razloga se, posebno kod Veb dizajna, sve više prelazi na vektorski definisane ikone, a *Scalable Vector Graphics* (SVG) jedan je od najpopularnijih formata.

10.1.2.1.2. Vremenska rezolucija

Jedna od osnovnih karakteristika grafičkog adaptera i displeja jeste brzina (učestalost, frekvencija) prikazivanja, odnosno izmene prikazanog grafičkog sadržaja. Kod grafičkog adaptera ova brzina izražava se u broju generisanih frejmova u sekundi (eng. *frames per second*, *FPS*), dok se kod displeja koristi broj osvežavanja prikaza u sekundi, odnosno učestalost koja se izražava u hercima. Broj frejmova u sekundi je promenljiv, dok je učestalost osvežavanja displeja podrazumevano konstantna (s tim da noviji monitori imaju određene mogućnosti prilagođavanja).

Broj frejmova u sekundi zavisi od više faktora. U slučaju da se prikazuje unapred definisan sadržaj (reprodukcija), maksimalan broj frejmova u sekundi biće prvenstveno ograničen samom vremenskom rezolucijom tog sadržaja. Na primer, nemi filmovi obično su snimani u 16 frejmova u sekundi, filmovi sa zvukom u 24, a standard za savremene digitalne video materijale je od 48 do 60 frejmova u sekundi. Snimljeni broj frejmova predstavlja najveći mogući izvorni kvalitet reprodukcije, a on može biti redukovano u slučaju da performanse samog računarskog uređaja ograničene. Nasuprot tome, danas je aktuelan i razvoj algoritama, najčešće zasnovanih na veštačkoj inteligenciji, koji mogu da generišu i umetnu dodatne frejmove u izvorni sadržaj.

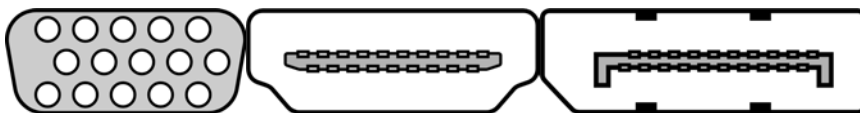
Nasuprot sadržaju koji se reprodukuje stoji sadržaj koji se generiše neposredno pred prikazivanje. Na primer, zadavanjem naredbi kretanja u 3D video igrama generiše se odgovarajuća vizuelna reprezentacija aktuelne scene, koja zavisi od nove pozicije igrača, ugla gledanja, stanja uključenih elemenata i ostalih igrača, i drugih elemenata. U slučaju da relevantne komponente računarskog sistema (prvenstveno centralni i grafički procesor) imaju visoke performanse, biće moguće generisanje velikog broja frejmova (na primer 60) koji opisuju jedan sekund navedenog kretanja. Žargonski rečeno, sa aspekta korisnika proces kretanja će izgledati "glatko". U suprotnom, ukoliko navedene komponente nemaju potrebne mogućnosti, broj frejmova će biti niži i prikaz kretanja će izgledati isprekidano.

Učestalost osvežavanja displeja takođe zavisi od više faktora. Prvenstveno, sam displej ima fizičko ograničenje u vidu najveće podržane učestalosti osvežavanja. Kod većine displeja ona iznosi 60Hz, s tim da kvalitetniji monitori

namenjeni za video igre podržavaju i učestalosti od 120, 144, pa čak i 240 herca.

Veoma važnu komponentu grafičkog sistema predstavlja i način povezivanja grafičkog adaptera i displeja. Uzmimo za primer situaciju u kojoj se koristi 4K monitor sa osvežavanjem od 120Hz, sa 24-bitnom (RGB) grafikom. Za maksimalne performanse ovakvog monitora potrebno je 120 puta u svakoj sekundi dostaviti informacije (po 24 bita) za svaki od preko 8 miliona (3840x2160) piksela. To ukupno čini preko 22Gb/s protoka podataka između grafičkog adaptera i displeja. Dakle, da bi se navedeni protok ostvario, potrebno je izabrati odgovarajući način povezivanja. U protivnom, sam sistem povezivanja može predstavljati “usko grlo”, odnosno onemogućiti postizanje maksimalnih performansi.

Tri najpopularnija interfejsa za povezivanje grafičkog adaptera i displeja čine VGA, HDMI, i DisplayPort. Interfejs pod nazivom *Video Graphics Array* (VGA) uveden je od strane IBM-a 1987. godine i omogućava maksimalnu rezoluciju od 2048x1536 piksela, sa osvežavanjem od 85Hz. *High-Definition Multimedia Interface* (HDMI) je vlasnički interfejs predstavljen 2002. godine, sa ciljem omogućavanja prenosa zvuka i većih rezolucija sa većim učestalostima osvežavanja, a trenutno poslednja verzija (2.1) nudi protok od 48Gb/s. *DisplayPort* je interfejs predstavljen 2006. godine od strane *Video Electronics Standards Association* (VESA) a poslednja verzija (2.0) omogućava protok od 80Gb/s između grafičkog adaptera i displeja. Svi navedeni standardi podržavaju EDID (eng. *Extended Display Identification Data*), odnosno standard kojim displej informiše grafički adapter o svojim mogućnostima - podržanim prostornim i vremenskim rezolucijama, načinima sinhronizacije, i drugi.



Slika 127. Izgled VGA, HDMI i DisplayPort konektora.

Način povezivanja displeja posebno je značajan kod prenosivih računara, pošto oni imaju ograničen broj portova i proširivost. Sa obzirom na to da se eksterni monitori u kombinaciji sa prenosivim računarima uglavnom koriste u kancelariji, čest način za njihovo povezivanje je preko doking stanice (eng. *docking station*). Trenutno najveći protok (40Gb/s) između doking stanice i računara može se ostvariti korišćenjem Thunderbolt 3/4, odnosno USB4 standarda. Međutim, ovaj protok nije dovoljan za optimalno korišćenje

monitora sa visokim rezolucijama (na primer 16K) i višim učestalostima osvežavanja.

10.1.2.1.3. Vremenska sinhronizacija

Da bi prikaz grafičkog sadržaja na displeju bio ispravan potrebno je da su frekvencija kojom od grafičkog adaptera stižu frejmovi i frekvencija osvežavanja samog displeja sinhronizovane. U protivnom, doći će do delimičnog preklapanja prikaza, odnosno displej će započeti sa prikazivanjem jednog frejma a u trenutku pristizanja novog frejma nastaviti sa njegovim prikazivanjem. Ovaj problem poznat je pod nazivom “cepanje ekrana” (eng. *screen tearing*). Posebno je izražen u situacijama kada grafički adapter ima značajno veću brzinu generisanja frejmova od brzine osvežavanja displeja.



Slika 128. Primer “cepanja ekrana”, odnosno kombinovanog prikazivanja dva frejma zbog nedostatka vremenske sinhronizacije.

Postoji više pristupa za rešavanje navedenog problema. Tehnologija pod nazivom *Vsync* funkcioniše po principu ograničavanja broja frejmova koje grafički adapter generiše na učestalost osvežavanja displeja. Nedostatak ovakvog pristupa je što se uzorkovanjem sa statičnom učestalošću ne dobijaju optimalni rezultati, odnosno može doći do preskakanja frejmova i različitog trajanja prikazivanja različitih frejmova. Za rešavanje ovog problema kompanija *AMD* je na svojim grafičkim adapterima podržala tehnologiju pod nazivom *FreeSync*, koja sinhronizuje učestalost osvežavanja displeja sa aktuelnom učestalošću generisanja frejmova od strane grafičkog adaptera. Kompanija *Nvidia* razvila je tehnologiju pod nazivom *G-Sync* koja takođe sinhronizuje učestalost prikazivanja frejmova na displeju sa učestalošću generisanja frejmova od strane grafičkog adaptera. Nedostatak ove tehnologije je to što se zasniva na zatvorenim komponentama, odnosno zahteva postojanje ugrađenog specifičnog *Nvidia* modula u sam displej.

10.1.2.2. Virtuelni grafički korisnički interfejs

Interfejsi ka korisničkom softveru se dele u dve osnovne kategorije: korisnički interfejs, namenjen ljudima, i programski interfejs (eng. *Application Programming Interface, API*) namenjen drugim programima. Međutim, postoje situacije u kojima je potrebno koristiti korisnički interfejs namenjen ljudima od

strane programa. Na primer, u situacijama kada je potrebno automatizovati testiranje korisničkog interfejsa ili određene funkcije aplikacije, moguće je koristiti namenske softverske alate (na primer *Selenium*) koji omogućavaju simuliranje korišćenja aplikacije od strane korisnika po zadatim instrukcijama, odnosno scenariju. Na taj način se ubrzava testiranje, testira efikasnost u radu sa velikim brojem paralelnih korisnika i štedi na testerima.

Jedan od glavnih nedostataka klasičnog korišćenja ovih alata je to što oni zahtevaju postojanje GKI, čime je broj paralelnih instanci testiranja ograničen na broj fizičkih korisničkih interfejsa na računaru, podrazumevano jedan. Međutim, postoje pristupi koji omogućavaju prevazilaženje ovog problema. Na primer, *Xvfb* je varijanta *X* servera na Linuks sistemima koja kreira virtuelni grafički korisnički interfejs, odnosno interfejs koji nije zaista povezan sa ulazno-izlaznim uređajima računara. Na taj način je omogućeno kreiranje većeg broja virtuelnih korisničkih interfejsa na jednom računaru, odnosno paralelno pokretanje većeg broja instanci testiranja. Obično se kod pokretanja nove instance virtuelnog interfejsa odmah zadaje zadatak koji će se preko njega izvršiti, a eventualnim rezultatima pristupa preko spoljne memorije (na primer kreiranjem i čuvanjem snimaka ekrana ili preuzetih fajlova u fajl sistemu).

Postoje i zloupotrebe ovakvih alata. Na primer, danas je na Vebu popularna kupovina “pažnje” na društvenim mrežama, odnosno postoje agencije koje za određenu novčanu nadoknadu klijentima pružaju određen broj “lajkova”, deljenja sadržaja i slično. Međutim, u većini slučajeva je, zapravo, kupljena pažnja nepostojećih korisnika, odnosno virtuelnih korisničkih profila. Iza takvih profila najčešće stoje sistemi koje funkcionišu (između ostalog) korišćenjem opisanih virtuelnih korisničkih interfejsa. Kompanije tvorcii društvenih mreža ulažu velike napore u otkrivanje i eliminisanje ovakvih profila.

10.2. Elementi GKI

Elementi savremenih GKI se mogu klasifikovati prema glavnoj funkciji koju obavljaju kao:

- Ulazni elementi,
- Navigacioni elementi,
- Informativni elementi,
- Kontejnerski elementi.

Ulazni elementi omogućavaju prosleđivanje informacija aplikaciji u obliku komandi i podataka. Navigacioni elementi omogućavaju kretanje kroz aplikaciju. Informativni elementi obezbeđuju povratnu informaciju ka korisniku. Kontejnerski elementi služe za organizaciju sadržaja interefejsa.

U nastavku je dat detaljniji opisa elemenata GKI.

10.2.1. Ulazni elementi

Jedan od osnovnih zadataka korisničkog interfejsa, pored prikazivanja podataka korisniku, čini preuzimanje podataka od korisnika. U suštini, svaka interakcija koja uključuje korisnika - na primer, pomeranje kursora miša ili klik na link na Veb stranici - se može posmatrati i kao prukupljanje podataka korisnika.

Forma je logička grupa elemenata (dugmad, tekstualna polja i drugi) za prosleđivanje podataka vezanih za specifičan zadatak. Sa obzirom da su pojedinačni elementi forme opisani zasebno, ovde se navode smernice za njihov dizajn [87]:

- Forme treba da budu što je moguće kraće (podatak koji se može dobiti kombinovanjem drugih podataka forme treba ukloniti).
- Grupisati labela i tekstualna polja tako da labela bude blizu polja na koje se odnosi.
- Organizovati elemente forme u jednoj koloni (više kolona otežava skeniranje).
- Elemente birati i prilagoditi tipu i veličini podataka koji se unose.
- Razlikovati i naznačiti obavezne i opcione elemente (opcion element je kandidat za eliminaciju iz forme i ne treba imati više od jednog do dva takva elementa).
- Ne koristiti opcije koje brišu tekući sadržaj forme (eng. *reset* i *clear*). Koristiti *cancel* opciju i korisniku prikazati dijalog potvrde koji govori da će podaci biti obrisani.
- Poruke o greškama unosa podataka moraju biti vidljive i odnositi se na dati element.

U osnovne ulazne elemente (koji se koriste i u formama) spadaju:

- tekstualno polje (eng. *text field*),
- dugme za čekiranje (eng. *checkbox*),
- naizmenično dugme (eng. *toggle button*),
- radio dugme (eng. *radio button*),
- dugme,
- padajuća lista (eng. *dropdown list*),
- klizač.

Jedan od najčešće korišćenih ulaznih elemenata jesu oni namenjeni direktnom unosu teksta. Postoji više varijanti ovih elemenata, i oni se grubo mogu podeliti

u one namenjene za unos kraćeg teksta (u jednom redu) i one za unos dužeg teksta u više redova (eng. *text area*).

Корисничко име:

Лозинка:

Једно од поља са најбржим
развојем телекомуникационих
технологија и сервиса јесте
Интернет мрежа. Иако
припадају релативно младој
телекомуникацији

Слика 129. Различити типови елемената за unos текста.

Значајан подтип елемената за unos текста u jednom redu jesu i polja za unos lozinki, kod kojih se ne prikazuje sadržaj koji je korisnik uneo. Takođe, različite biblioteke i okviri nude različite mogućnosti za ograničavanje unosa, na primer na unos samo brojeva, sa ograničenom dužinom teksta ili korišćenjem regularnih izraza.

Još jedna situacija koja podrazumeva relativno složen unos od strane korisnika jeste unos datuma i vremena. Obično korisnik prilikom izbora datuma želi da proveru u kalendaru koji je dan u pitanju, a potrebno je i da se datum i vreme unesu u ispravnom formatu (redosled elemenata, separator i format). Da bi se olakšao rad korisnicima, kao i smanjile greške prilikom unosa, često se ručni unos zamenjuje naprednijim kontrolama koje omogućavaju prikazivanje kalendara, odnosno izbor datuma i vremena.

2020/10/19 20:42

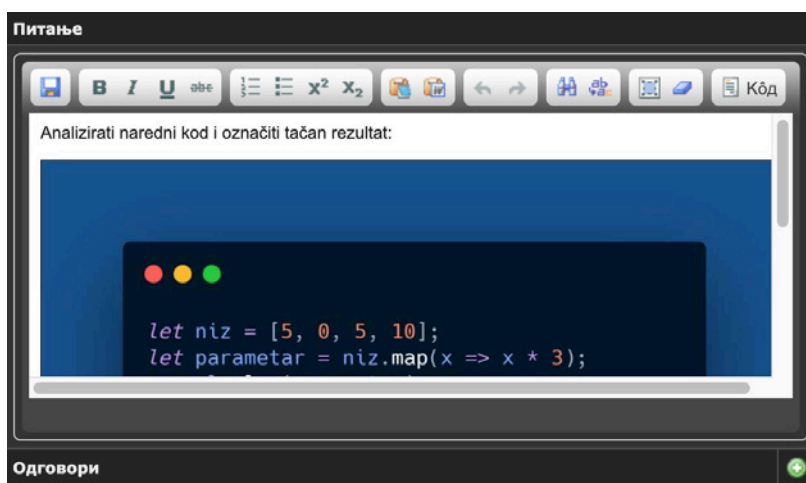
October 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

18:42
19:42
20:42
21:42
22:42
23:42

Слика 130. Пример улазног текстуалног елемента за unos datuma i vremena pomoću kalendara.

Jedan od osnovnih nedostataka kontrola za unos teksta u više redova je to što one podrazumevano ne podržavaju formatiranje teksta, kao ni dodavanje grafičkog sadržaja. Iz tog razloga se u praksi, u Veb aplikacijama, koriste dodatne biblioteke koje ove kontrole dopunjuju navedenim funkcionalnostima. Naziv za ovakve biblioteke je WYSIWYG (eng. *What You See Is What You Get*), inspirisan činjenicom da se formatiranje direktno prikazuje, a ne unosi u vidu dopunskih oznaka (na primer *HTML* i *LaTeX*).



Slika 131. Primer korišćenja WYSIWYG editora kao dopune kontrole za unos teksta u više redova.

Osim slobodnog unosa, u GKI se često koriste i elementi kod kojih korisnik može da izabere jednu ili više ponuđenih opcija. Prednost ovakvog unosa je brzina i standardizacija unosa, i eliminacija grešaka. Kod izbora ponuđenih vrednosti ulazni elementi se u osnovi mogu podeliti po tome da li omogućavaju izbor samo jedne ili više ponuđenih vrednosti. Za izbor jedne vrednosti uglavnom se koriste radio dugmad i meniji. Prednost radio dugmadi je što se odmah vide sve ponuđene opcije, dok se u menijima štedi prostor na korisničkom interfejsu (po cenu dodatnog klika za pristup vrednostima). Za izbor više ponuđenih vrednosti uglavnom se koriste dugmad za čekiranje i liste (Slika 132).

Интернету приступате:

- ☒ код куће
- ☐ на послу
- ☒ на факултету

Интернету приступате:

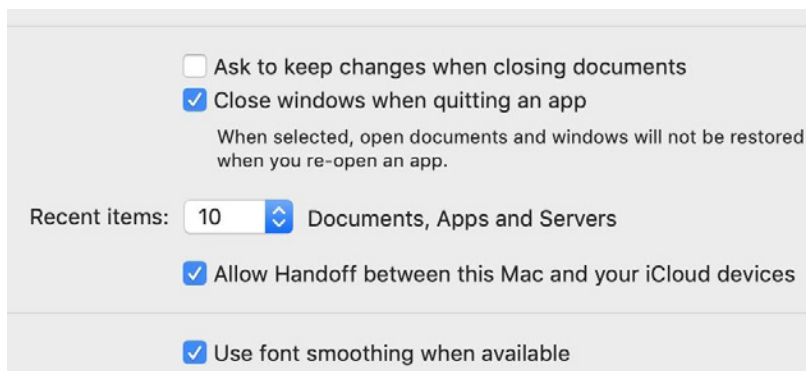
код куће

на послу

на факултету

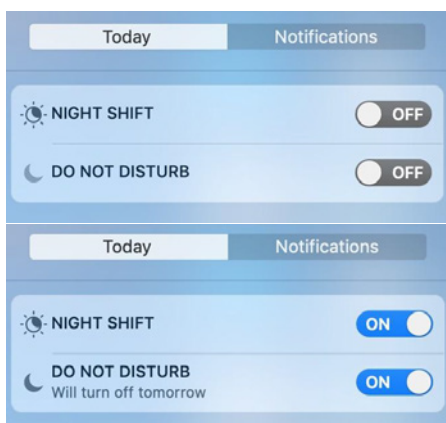
Slika 132. Primeri elemenata GKI za izbor većeg broja ponuđenih opcija.

Dugme za čekiranje omogućava korisniku da izabere jednu ili više opcija (kao dugmad) iz skupa opcija koje nisu međusobno isključive. Vertikalna lista dugmadi je lakša za vizuelno skeniranje i prepoznavanje njenog sadržaja (Slika 133). Namena elementa jesu specifična podešavanja (na primer, funkcije ili dela aplikacije) koja sadrže veći broj opcija koje se međusobno ne isključuju.



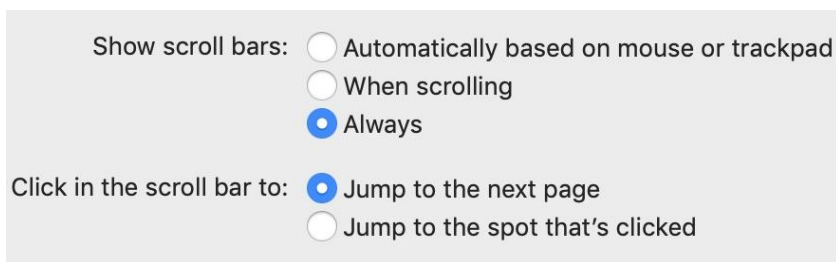
Slika 133. Dugme za čekiranje služi za podešavanja kod kojih opcije nisu međusobno isključive.

Naizmenično dugme je primer metafore iz realnog sveta. Omogućava korisniku promenu između dva stanja aplikacije koja su suprotna i međusobno isključiva. Mora postojati vizuelni efekat promene koji je neposredno vidljiv iz samog elementa kako bi izbor bio nedvosmislen. Drugim rečima, stanja se vizuelno diferenciraju (na primer različitim bojama dugmadi kao na slici ispod).



Slika 134. Naizmenično dugme (levo) je primer metafore iz realnog sveta (desno).

Radio dugme takođe predstavlja primer metafore iz realnog sveta (slika ispod). Omogućava korisniku da izabere jednu opciju iz skupa opcija (dugmadi) koje su međusobno isključive. Služi prevenciji grešaka - korisnik ne može uneti pogrešne podatke jer su ponuđene validne opcije. Vertikalna lista dugmadi je lakša za vizuelno skeniranje. Koriste se uglavnom za podešavanja parametara, pre nego kao komande za pokretanje akcija.



(a)



(b)

Slika 135. Radio dugme služi za izbor opcija koje međusobno isključive (a) i vodi poreklo iz realnog sveta (b).

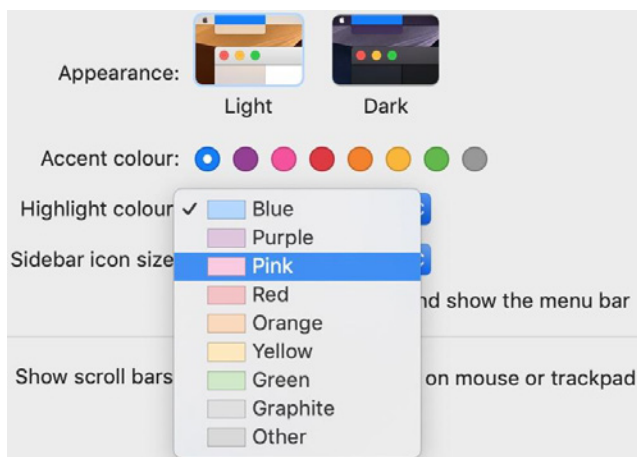
Tabela 7 daje uporednu analizu tri ulazna elementa GKI.

Tabela 7. Uporedni prikaz karakteristika radio dugmeta, dugmeta za čekiranje i naizmeničnog dugmeta.

Karakteristika	Radio dugme	Dugme za čekiranje	Naizmenično dugme
<i>Broj opcija</i>	>1	>1	1
<i>Broj selekcija od strane korisnika</i>	1	0-N	2
<i>Podrazumevana opcija</i>	Da	Ne	Da
<i>Odnos opcija</i>	Međusobno isključive	Nezavisne	Međusobno isključive
<i>Efekat selekcije</i>	Kada korisnik izabere opciju za potvrdu	Kada korisnik izabere opciju za potvrdu	Neposredno

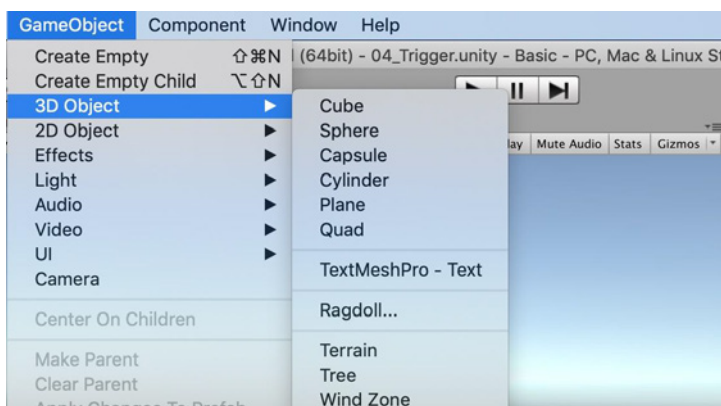
Dugme označava akciju koja je vizuelno opisana labelom (nazivom), simbolom ili kombinacijom dva elementa. Dugme opisano simbolom je poznatije kao ikona. Mora biti vidljivo na interfejsu (dobar kontrast u odnosu na susedne elemente) i obezbediti vizuelni efekat pokretanja akcije (klika dugmeta). Koristiti standardne oblike za dugme (varijante pravougaonika). Naziv mora biti kratak, razumljiv i asocirati na akciju koju dugme pokreće. Slično tome, simbol ikone treba izabrati pažljivo tako da značenje podseća na akciju koju dugme inicira.

Padajuća lista je lista koja omogućava korisnicima izbor jedne opcije u datom trenutku. Slično radio dugmetu, ali sa većim brojem opcija koje se prikazuju na zahtev kako ne bi zauzimale prostor na interfejsu. Opcije mogu biti komande, linkovi za navigaciju, i skupovi elemenata ili atributa. Naziv padajuće liste uvek mora biti vidljiv.



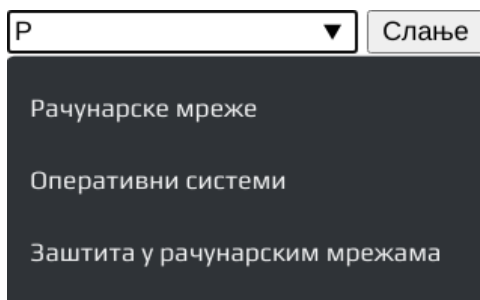
Slika 136. Padajuća lista daje veći broj opcija od kojih se jedna može izabrati u trenutku. Naziv liste mora biti vidljiv i kada je lista prikazana.

Sadržaj jedne liste mora biti konzistentan u pogledu opcija bez obzira na različite kontekste izvršavanja. Treba izbegavati promene sadržaja, odnosno dodavanje ili elimisanje opcija u zavisnosti od stanja aplikacije. Umesto toga, opcije koje ne bi trebalo da budu dostupne u datom režimu rada se mogu učiniti neaktivnim (eng. *disabled*). Pored toga, ne treba imati dugačke padajuće liste čija je visina veća od visine ekrana. Ovo može smanjiti efikasnost ukoliko korisnik mora da skroluje vertikalno da bi našao željenu opciju. U tom slučaju koristiti kaskadne liste (eng. *cascading lists*).



Slika 137. Padajuća lista mora imati isti sadržaj za različite kontekste aplikacije. Opcije nedostupne za dati kontekst se mogu učiniti neaktivnim. U slučaju većeg broja opcija lista može biti kaskadna.

Kombinaciju slobodnog unosa i izbora iz ponuđenih vrednosti predstavljaju takozvane liste podataka (eng. *datalist*). Prednost ovih elemenata je fleksibilnost - nude mogućnost izbora, ali ne ograničavaju korisnički unos samo na predefinisane vrednosti. Ponuđeni izbor se sužava u toku samog unosa korisnika, tako da se i smanjuje broj sličnih vrednosti, odnosno povećava uniformnost unosa.



Slika 138. Liste podataka kombinuju slobodan unos i ponuđene stavke.

Slično ponašanje nudi i *autocomplete* funkcionalnost u Veb pretraživačima, sa tim da postoje ključne razlike - stavke u listi podataka definišu se u samom elementu GKI, odnosno ne zavise od klijenta. Takođe, uklanjaju se i bezbednosni rizici vezani za situacije kada više osoba koristi isti Veb pretraživač.

10.2.2. Navigacioni elementi

Intuitivno i efikasno kretanje kroz određeni sadržaj, na primer Veb stranicu ili korisničku aplikaciju, jedan je od osnovnih zadataka dizajna korisničkog interfejsa. Ovaj zadatak je složen i zavisi od više faktora. Kao prvo, sam sadržaj mora biti tako organizovan da korisnik intuitivno može zaključiti gde se željeni podskup informacija nalazi, odnosno kojim navigacionim koracima će do njega stići. Sa druge strane, potrebno je opcije za navigaciju definisati odgovarajućim elementima koji, ne samo da pružaju potrebnu funkcionalnost i efikasnost, već su i sami dovoljno intuitivni da ih korisnik može bez napora koristiti.

U skladu sa ovim kriterijuma u praksi se izdvojio relativno mali skup navigacionih elemenata koji se standardno koriste za potrebe kretanja korisnika kroz GKI. U osnovne navigacione elemente GKI spadaju:

- putanja,
- kartica (eng. *tab*),
- meni,
- straničenje.

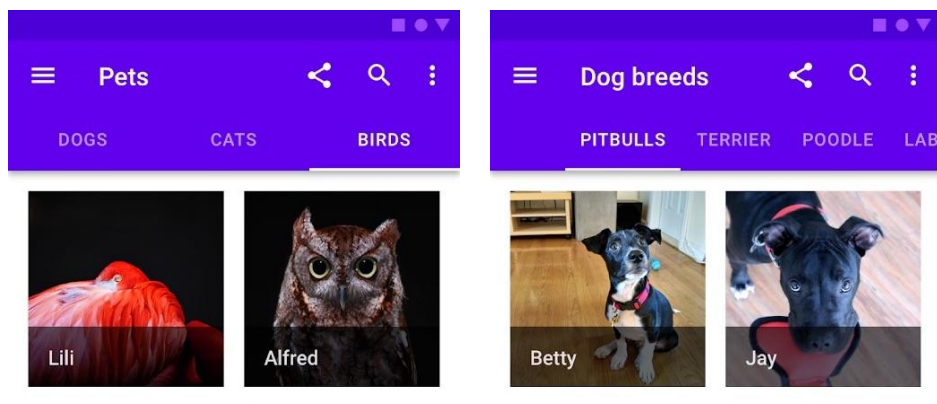
Putanja predstavlja liniju teksta i simbola koja pokazuje trenutnu lokaciju (stranicu) u hijerarhiji interfejsa. Zasnovana je na metafori čitanja kao redosleda procesiranja sadržaja sa leve na desnu stranu (za određene jezike). Jedna od najpopularnijih kontrola ovog tipa je ona pod nazivom “mrvice hleba” (eng. *breadcrumbs*). Ova komponenta je dobila naziv po strategiji Ivice i Marice, junaka iz čuvene bajke braće Grim, da bacanjem mrvica hleba označe put kako bi mogli da se vrate iz šume.

<https://hcmmap.online> » Content » Page » MT Experiment Tallinn » Modules

Slika 139. Primer korišćenja “mrvice hleba” kao elementa navigacije u Veb aplikaciji.

Osim što imaju zadatak da omoguće povratak korisnika na prethodne ili početnu stranicu, “mrvice hleba” su značajne i zbog toga što korisniku daju informaciju gde se trenutno nalazi [50] - ne samo u pogledu naziva aktuelne stranice, već i logičke putanje na kojoj se ona nalazi. Element čine uključene stranice putanje po strukturi sadržaja, pre nego po hronološkom redu njihovog posećivanja. Takođe, nije nužno da svi elementi putanje budu prikazani, u zavisnosti od njihove važnosti i raspoloživog prostora za prikazivanje. Zbog značaja koji ovaj element ima, on se postavlja na vidljivu poziciju, u zaglavlje Veb stranice.

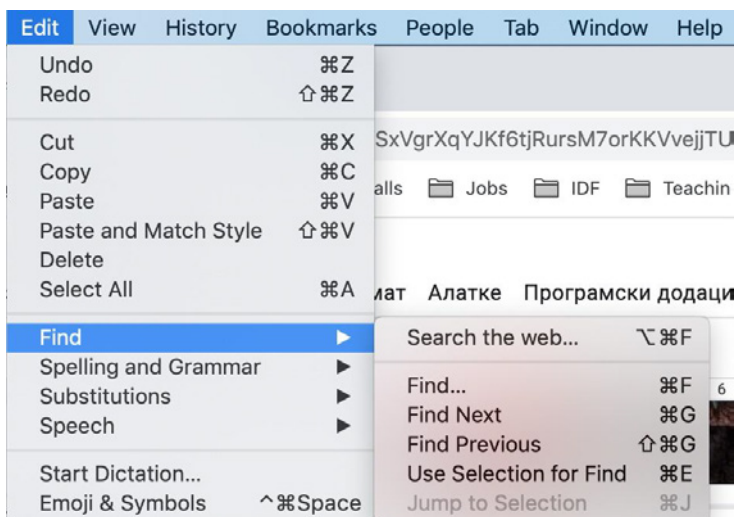
Kartica je navigacioni element sekundarnog nivoa. Za razliku od elementa putanje koji omogućava navigaciju između delova GKI sa drugačijim sadržajem, kartica služi za kretanje između delova koji su logički povezani, odnosno predstavljaju delove jedne celine. Kaže se da delovi imaju isti kontekst sadržaja određen celinom (kao na slici ispod). Svaka kartica predstavlja orijentir panela (eng. *panel*) sa sadržajem. Kod ovog elementa važna je vizuelna povratna informacija da je kartica aktivna. Naziv kartice mora biti vidljiv, kratak, razumljiv i asocirati na sadržaj. Kartice uvek rasporediti u jedan red (u suprotnom je otežano vizuelno skeniranje). Liniju kartica treba smestiti pri vrhu stranice. Pored toga, potrebno je obezbediti konzistentan izgled kartica i njihovih panela (boje, elementi, raspored i drugi), nezavisno od razlike u sadržajima.



Slika 140. Kartica je navigacioni element sekundarnog nivoa koji grupiše i prikazuje sadržaje u okviru jednog konteksta ili celine.

Izvor: <https://material.io/components/tabs> (15.11.2020).

Meni je element koji opisuje grupu komandi ili primeraka kategorije odvojenih od ostatka dizajna koji se prikazuju na zahtev. Pripada kategoriji navigacionih elemenata zbot toga što omogućava kretanje kroz različite tipove sadržaja poput naredbi akcija, delova interfejsa, objekata interfejsa i drugih. Kod menija navigacija ima šire značenje u pogledu sadržaja kroz koji se korisnik može kretati. Zbog toga i predstavlja najzastupljeniji navigacioni element GKI. Meniji se nalaze na lokacijama na kojima ih korisnik očekuje (gornji deo ili leva traka interfejsa). Stavke menija moraju izgledati interaktivno, odnosno treba obezbediti dobar kontrast boje teksta i boje pozadine stavki menija. Naziv menija mora uvek biti vidljiv. Nazivi stavki moraju asocirati na funkcije ili sadržaje na koje upućuju. Menije i stavke ravnati po levoj strani radi lakšeg skeniranja. Veličina menija (kao broj stavki) ne sme biti takva da zahteva vertikalno skrolovanje. U tom slučaju koristiti kaskadne menije. Dubina kao vertikalna dimenzija sadrži kategorije, dok širina kao horizontalna dimenzija obuhvata podkategorije. Radi lakšeg i bržeg korišćenja preporuka je imati do dva nivoa širine kaskadnih menija (kao na slici u nastavku).



Slika 141. Meni grupiše i prikazuje akcije na zahtev. Redosled akcija obično određuje relevantnost u pogledu učestalosti pozivanja. U slučaju većeg broja mogućih akcija mogu se koristiti kaskadni meniji.

Posebnu vrstu menija predstavljaju takozvani “hamburger meniji”, odnosno meniji koji su na korisničkom interfejsu predstavljeni ikonu koja vizuelno podseća na hamburger, a sadržaj menija se prikazuje klikom na ikonu. Razlog za njihovo korišćenje je ograničen prostor na mobilnim interfejsima. Na primer, ovakvi meniji postali su standard kod prilagodljivog dizajna Veb sajtova (eng. *responsive design*) tako da se skup opcija kod stonih klijenata prikazuje u zaglavlju interfejsa, dok se isti skup nalazi u meniju kod prikaza na mobilnim uređajima.



Slika 142. Standardan izgled simbola ikone za oznaku “hamburger” menija.

Straničenje se koristi u situacijama kada je broj elemenata (stavki ili rezultata) preveliki da bi se prikazao na jednoj stranici. Pod straničenjem se podrazumeva podela rezultata u više stranica sa ograničenim brojem prikazanih stavki i omogućavanje navigacije između stranica. Pri tome se mora voditi računa da navigacija po stranicama bude efikasna, odnosno da korisnik lako (sa najviše dva klika, a poželjno jednim) dođe do rezultata/segmenta koji mu je potreban. Treba imati u vidu da se razlaganjem rezultata na više stranica kod Veb GKI efikasnost korišćenja može smanjiti u slučaju dobavljanja stranica na zahtev (u

zavisnosti od vrste i količine podataka). Standardna je komponenta tabele kao elementa GKI (slika u nastavku).

ID	▼ Name				
Search by ID		Search by Name			
1	001/00000001	00000001 - 00000001			
2	001/00000002	00000002 - 00000002			
3	001/00000003	00000003 - 00000003			
4	001/00000004	00000004 - 00000004			
5	001/00000005	00000005 - 00000005			
  5  1 - 5 of 46 1 of 10   					

Slika 143. Straničenje podataka u tabeli, sa mogućnošću izbora broja rezultata po stranici, informacijom o trenutnoj stranici, mogućnošću skoka na određenu stranicu i navigacijom na prvu/prethodnu/sledeću/poslednju stranicu.

Često se koristi za kretanje kroz rezultate pretrage. Kod krajnjih korisnika kreira doživljaj kontrole zbog uvida u veličinu kolekcije elemenata, trenutnu poziciju u kolekciji, i mogućnosti kretanja kroz kolekciju (videti Poglavlje 9.1.8, višestruke uloge).

A Systematic Review of Artificial Intelligence and Robots in ...

Oct 17, 2018 — This **systems-oriented** conceptualization captures the nature of value through four ... This **study** employs a **systematic literature review** as its methodology. ... Journal of Marketing Research, Journal of Consumer Research, Marketing ... Third, whenever content related to AI or robots in value co-creation was ...

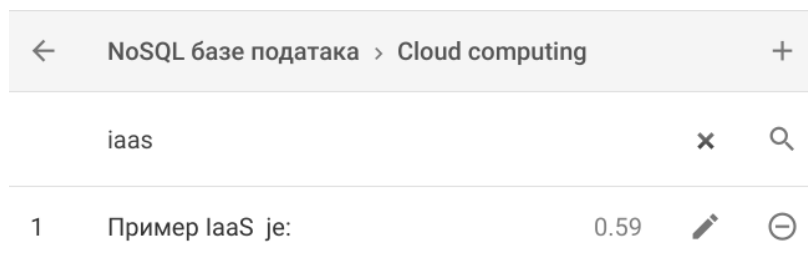
by V Kaartemo · 2018 · Cited by 22 · Related articles



Slika 144. Straničenje se koristi za prikaz rezultata pretrage ukoliko se oni ne mogu prikazati na jednoj strani interfejsa.

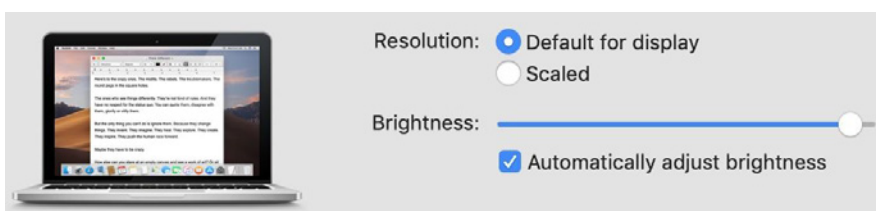
Kada korisnik pristupa sadržaju u vidu velikog broja objekata koji nisu precizno kategorisani (na primer, tekstovi na blogu ili društvenim mrežama), odnosno mogu pripadati većem broju kategorija, jedan od pristupa za olakšavanje navigacije je korišćenje oznaka (eng. *tags*). Oznake su kratki (najčešće jedna reč) opisi objekata koji se sa semantičkog značenja mogu pozicionirati između kategorija i ključnih reči. Njihova uloga je da korisnik može lako pronaći srodne tekstove, odnosno tekstove koji su takođe označeni izabranom oznakom.

U slučajevima kada je broj mogućih opcija preveliki, odnosno kada ni jedna od opisanih tehnika nije u stanju da omogući efikasan pristup željenim informacijama, pretraga na osnovu upita korisnika može biti optimalno rešenje. Osnovni preduslov za uspešno pretraživanje jeste pravilno upoređivanje pojma za koji se vrši pretraga sa postojećim podacima. Kod složenijih pretraga je važno omogućiti rangiranje (sortiranje) rezultata pretrage. Obično korisnik bira kriterijum po kom želi da sortira rezultate (na primer, od novijih ka starijim podacima ili od jeftinijih ka skupljim).

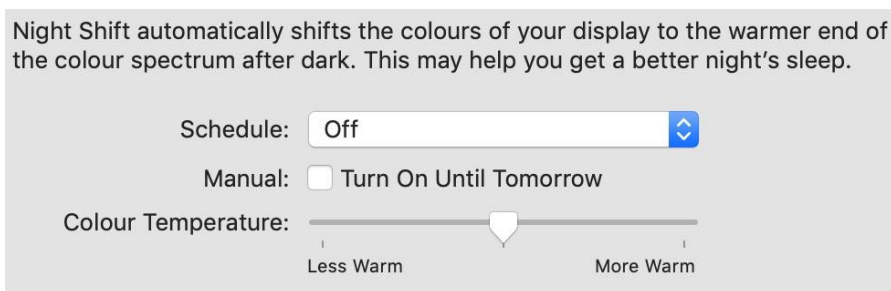


Slika 145. Primer korišćenja pretrage za izdvajanje (filtriranje) željenih informacija.

Klizač je ulazni element kod kojeg se prevlačenjem ručice po pravcu skale definiše vrednost varijable iz definisanog opsega. Varijabla predstavlja specifičan parametar u zavisnosti od konteksta korišćenja, na primer kontrola signala (zvuk) ili filtriranje podataka (pretraga). Kritičan aspekt klizača je preciznost kao nivo kontrole kod zadavanja konkretne vrednosti. U tom pogledu klizač može biti kontinualan (eng. *continuous*) ili diskretan (eng. *discrete*) kao što je prikazano na slici u nastavku. Kontinualan klizač omogućava izbor proizvoljne vrednosti iz opsega, dok se kod diskretnog klizača mogu izabrati predefinisane vrednosti. Kod dodirnih interfejsa treba voditi računa da svaka labela klizača mora biti pored ili iznad prsta korisnika i ručice kako bi bili vidljivi u toku prevlačenja.



(a)



(b)

Slika 146. Varijante klizača kao (a) kontinualan i (b) diskretan.

10.2.3. Informativni elementi

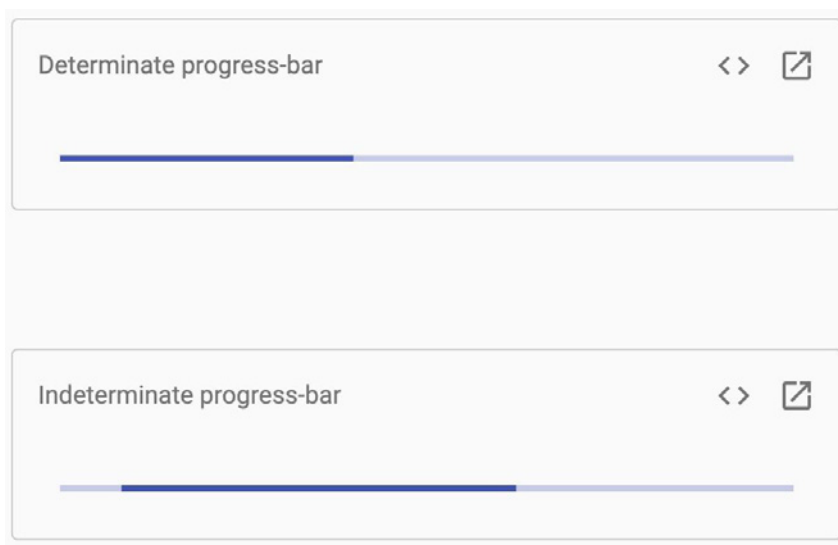
Kao informativni elementi GKI pojavljuju se:

- progres (eng. *progress*) i
- dijalog (eng. *dialog box*).

Progres je element koji informiše korisnika o stanju procesa kao trajanja pokrenute akcije. Povratna informacija korisniku da izvršenje akcije traje određeno vreme redukuje doživljaje neodređenosti i nezadovoljstva kod

korisnika. Na ovaj način se korisnicima daje do znanja da sistem “uvažava” njihovo vreme. Pored toga, redukuje se percepcija vremena korisnika zbog toga što je njihova pažnja usmerena na element, a ne na protok vremena. Povratna informacija mora biti trenutna jer vreme čekanja korisnika počinje onog trenutka kada inicira akciju. Progres koristiti za svaku akciju koja traje duže od nekoliko sekundi. Isto tako, koristiti dinamičke elemente kao animacije, a ne statičke poruke (labele) o progresu.

Postoje dve vrste progresa: određeni (eng. *determinate*) i neodređeni (eng. *indeterminate*) [88]. Neodređeni progres ne daje precizniju informaciju u vezi sa trajanjem pokrenute akcije i koristi se kod akcija čije vreme izvršavanja ne može biti unapred utvrđeno, odnosno može varirati. Može se koristiti za akcije kod kojih se može očekivati trajanje do 10 sekundi. Određeni progres daje precizniju informaciju o trajanju pokrenute akcije i kvantifikuje udeo izvršenja akcije. Javlja se kao linearan ili cirkularan, a ponekad ima i brojčanu vrednost (kao trenutni procenat izvršenja akcije). Može se koristiti za akcije koje mogu trajati duže od 10 sekundi.



Slika 147. Određeni i neodređeni progres elementi GKI.

Izvor: <https://material.angular.io/components/progress-bar/examples> (15.11.2020).

Dijalog potvrde je element koji sumira akciju koju korisnici žele da izvrše i traži od korisnika potvrdu izvršenja. Predstavlja mehanizam prevencije grešaka (videti Poglavlje 3.3.2). Često bude modalan - istaknut i u fokusu dok je ostatak

prikaza neaktivan kako bi se korisniku dodatno privukla pažnja. Sa druge strane, dijalog predstavlja prekid prirodnog toka akcija i razmišljanja korisnika, i zbog toga zahteva pažljivu primenu. Pre svega da bi se smanjio rizik da korisnik automatski potvrdi upozorenje bez razumevanja posledica. U skladu sa tim element treba koristiti samo za akcije sa ozbiljnijim posledicama koje mogu uzrokovati gubitak podataka korisnika i prelazak sistema u stanje iz kojeg se ne može dalje koristiti. Ne treba ih koristiti za česte, rutinske akcije koje ne mogu imati ozbiljnije posledice po rad sistema. U suprotnom, korisnik prestaje da obraća pažnju i povećava se rizik iznad (analogija sa basnom o pastiru i vuku).

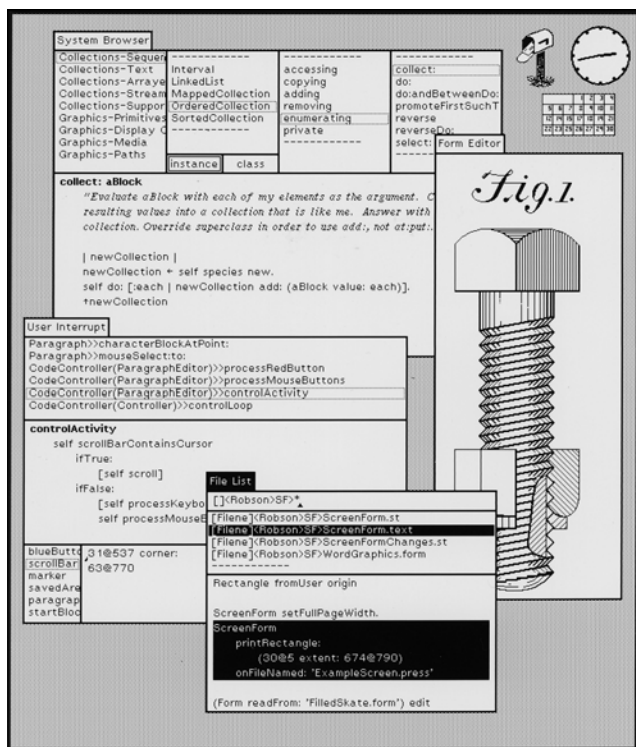
10.2.4. Kontejnerski elementi

Kontejnerski elementi GKI se mogu klasifikovati kao:

- prozor (eng. *window*),
- panel, i
- linija alata (eng. *toolbar*).

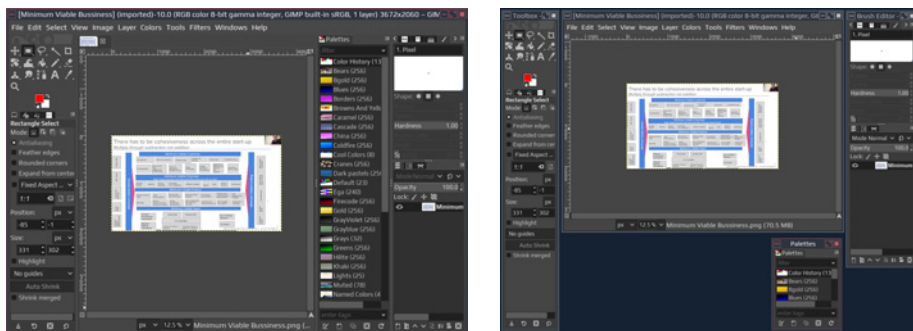
Prozor i panel su bazični kontejnerski elementi u GKI. U dizajnu ovih elemenata se primenjuju principi opisani u Poglavljima 1, 3 i 9.

Već u ranim fazama razvoja GKI prozori su se istakli kao podrazumevani element za organizaciju prikazanog sadržaja. Pojavom *multitasking* paradigme u GKI, problem koji se javio na nivou korisničkog interfejsa bio je kako jednostavno imati uvid u trenutno aktivne aplikacije, odnosno kako jednostavno prelaziti iz jedne u drugu. Pored različitih ostalih rešenja, uglavnom zasnovanih na korišćenju prečica na tastaturi, prozori su se, još i pre razvoja GKI, izdvojili kao intuitivno i efikasno rešenje. Kompanija Majkrosoft je iskoristila popularnost prozora tako da je svoj prvi operativni sistem koji je imao podršku za ovaj način interakcije sa korisnicima nazvala Prozori (eng. *Windows*).



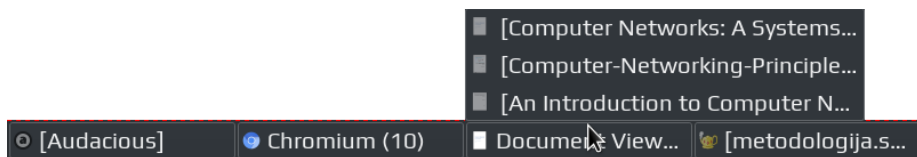
Slika 148. Jedan od prvih računara koji su imali GKI sa prozorima, Xerox Alto, razvijen je početkom 70-ih godina 20. veka.

Prve biblioteke za GKI podržavale su prikaz jednog prozora po aplikaciji, odnosno korišćenje prozora za razdvajanje različitih aplikacija koje se paralelno koriste na računaru. Međutim, sa porastom popularnosti prozora, omogućeno je i njihovo korišćenje unutar samih aplikacija, odnosno prikazivanje više prozora po jednoj aktivnoj aplikaciji. Obično su svi prozori aktivnih aplikacija registrovani i prikazani na liniji zadataka (eng. *taskbar*), tako da se određeni prozor može aktivirati ili smanjiti klikom miša na njegovu ikonu/naziv na ovoj liniji.



Slika 149. *Gimp*, softverski alat otvorenog koda za obradu rasterske grafike, ima mogućnost da sve elemente prikazuje unutar jednog prozora (levo) ili da glavne elemente i otvorene datoteke prikazuje u odvojenim prozorima (desno).

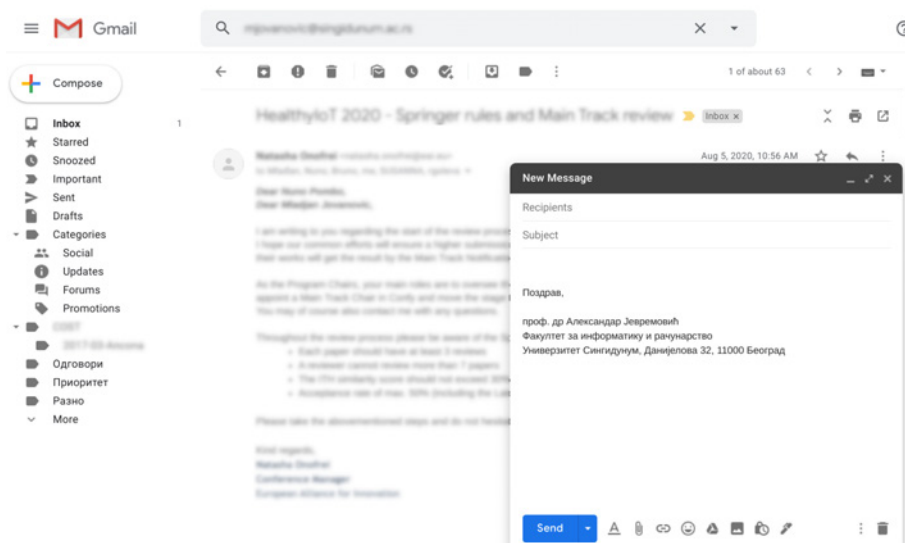
Jedan od značajnijih problema prozora, koji se javio sa rastom mogućnosti računarskih sistema, čini i snalaženje u situacijama kada je na računaru aktivan veliki broj aplikacija, odnosno pronalaženje željenog prozora. Postoji nekoliko pristupa za rešavanje ovog problema. U nekim operativnim sistemima, odnosno grafičkim okruženjima, moguće je uključiti grupisanje prozora koji pripadaju istoj aplikaciji. Nedostatak ovog pristupa su smanjena preglednost i potreba da se kod svakog pristupa prozoru iz grupe dva puta klikne mišem - jednom za otvaranje grupe, a zatim još jednom za izbor prozora iz grupe. Sa druge strane, kod većine popularnih Veb pretraživača preferira se korišćenje kartica odnosno izbegava se otvaranje novog prozora za svaku Veb stranicu koja se koristi.



Slika 150. Primer grupisanja prozora iste aplikacije na liniji zadataka u *XFCE* grafičkom korisničkom okruženju.

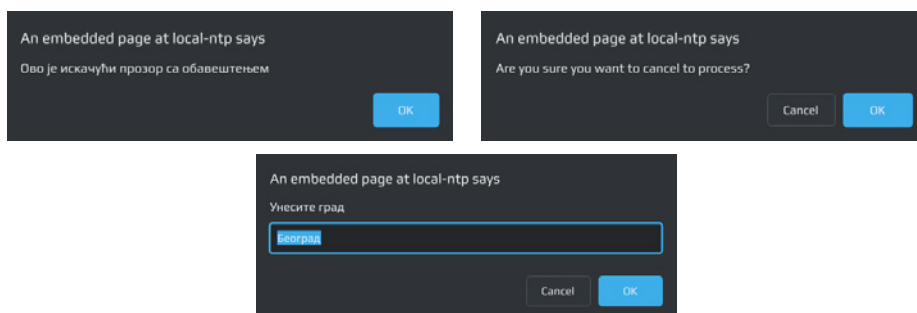
U slučaju složenih korisničkih interfejsa, segmentacija sadržaja korišćenjem zasebnih prozora može olakšati korisniku interakciju sa aplikacijom. Osnovni tip prozora u aplikaciji jesu obični, nemodalni prozori unutar glavnog prozora. Namena prozora ovog tipa je da se u njih izdvoji određen skup elemenata i sadržaja, najčešće namenjenih za izvršavanje određene akcije. S obzirom da ovakvi prozori ne sprečavaju pristup ostalim delovima korisničkog interfejsa, korisnik može da se kreće između ovakvih prozora i glavnog prozora, da im menja veličinu i smanjuje ih, kao i pojedinačno zatvara. Za razliku od iskačućih

prozora, nemodalni prozori su deo glavnog prozora, što znači da ne mogu da se pomeraju van njegovih granica.



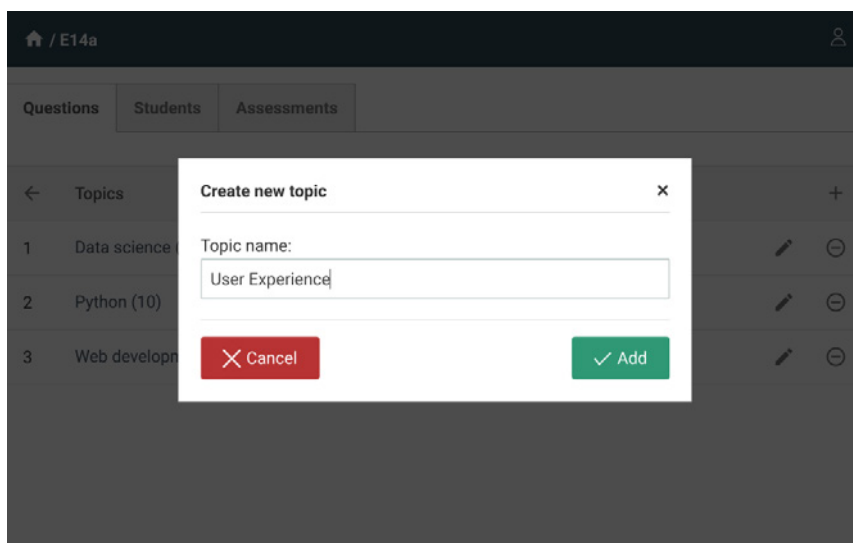
Slika 151. Primer korišćenja nemodalnog prozora, odnosno prozora koji ne blokira ostali sadržaj.

U situacijama u kojima je potrebno prekinuti podrazumevani tok korišćenja i pažnju korisnika usmeriti na određeno pitanje ili obaveštenje, iskačući dijalozi i modalni prozori često predstavljaju optimalan izbor komponente. Iskačući dijalozi mogu blokirati pristup glavnom sadržaju, ali se to ne podrazumeva. Ukoliko dijalog ne blokira pristup glavnom prozoru, može se desiti da ga korisnik slučajno previdi (na primer klikne na glavni prozor i pošalje iskačući prozor u pozadinu).



Slika 152. Iskačući dijalozi za obaveštavanje, dobijanje potvrde i unos vrednosti od strane korisnika, podrazumevano podržani u svim Veb pretraživačima koji imaju podršku za JavaScript.

Modalni prozori imaju sličnu namenu kao i iskačući dijalozi, sa tom razlikom da se obično koriste za složeniju interakciju sa korisnikom i da se razlikuju na tehničkom nivou. Za njihovo kreiranje se uglavnom ne koriste ugrađene funkcije Veb pretraživača, već složenije biblioteke koje nude znatno veću mogućnost prilagođavanja sadržaja i ponašanja potrebama korisnika. Modalni prozori predstavljaju sastavni deo glavne aplikacije, dok su iskačući prozori nezavisni. U skladu sa tim, modalni prozori (dok su otvoreni) blokiraju postojeći sadržaj korisničkog interfejsa, kako vizuelno, tako i funkcionalno. Zatim, za razliku od iskačućih prozora, aplikacija u jednom trenutku može da ima samo jedan modalni prozor.



Slika 153. Primer upotrebe modalnog dijaloga umesto preusmeravanja korisnika na novu stranicu sa formom.

Prilikom dizajniranja modalnih dijaloga treba voditi računa o primeni principa koji su se najbolje pokazali u praksi i koji se oslanjaju na postojeće korisničko iskustvo. Na primer, ukoliko je predviđena mogućnost da korisnik zatvori modalni dijalog, treba mu omogućiti da to učini na sve uobičajene načine: putem dugmeta (X) u gornjem desnom delu prozora, zatim putem dugmeta u okviru sadržaja prozora (na primer *Close*, *Cancel*), putem klika na zatamnjenju pozadinu oko dijaloga, kao i pritiskom tastera *Escape* na tastaturi. Zatim, usmeravanje pažnje korisnika prekidom toka akcija može uticati da korisnik zaboravi šta je bio osnovni tok, odnosno potreban je dodatni napor da se vrati u prethodni kontekst. Zbog toga se ne preporučuje korišćenje modalnih prozora unutar već otvorenog modalnog prozora.

Treba imati u vidu i to da modalni prozori nisu zamena za osnovne stranice, odnosno da najčešće nije opravdana realizacija svih stranica aplikacije u vidu modalnih prozora. U slučaju da je sadržaj modalnog prozora preveliki, odnosno zahteva skrolovanje da bi se u potpunosti prikazao, treba razmotriti opciju izmeštanja dijaloga u zasebnu stranicu. Takođe, u Veb okruženju, ka stranicama je moguće definisati jednoznačne linkove, dok povezivanje modalnih prozora najčešće nije moguće. Ukoliko su za interakciju sa modalnim prozorom potrebne dodatne informacije, koje se ne nalaze unutar samog prozora, treba razmotriti da li je njegova upotreba opravdana.

Linija alata je element koji predstavlja horizontalnu (ili vertikalnu) traku u koju se smeštaju ulazni elementi tipa dugme (varijanta ikona) koji pokreću odgovarajuće akcije (alate). Linije je potrebno smestiti u vidljivom delu interfejsa (pri vrhu ili na levoj strani). Alati moraju biti dovoljno istaknuti i razdvojeni radi prevencije grešaka kao izbora pogrešnog alata. Pored toga, potrebno je obezbediti veličinu alata koja omogućava pravilnu selekciju (naročito na mobilnom interfejsu).

10.3. Principi dizajna korisničkog interfejsa

Principi dizajna korisničkog interfejsa su generalizovane preporuke o tome kako korisnički interfejs treba da izgleda i funkcioniše. Cilj je unapređenje upotrebljivosti i prijatnosti korišćenja korisničkog interfejsa. Principu su zasnovani na psihologiji čoveka, odnosno načinima na koje opažamo, učimo, rezonujemo i pamtimo akcije nad korisničkim interfejsom. Kao takvi su univerzalni i relevantni za različite vrste korisničkih interfejsa (vizuelni, glasovni i drugi). Principe je ustanovio Ben Šnajderman 90-tih godina 20. veka [79]. Formulirani su kao osam zlatnih pravila (eng. *golden rules*) dizajna korisničkog interfejsa i najčešće se koriste u praksi. Pravila u dobroj meri podsećaju na heuristike upotrebljivosti opisane u Poglavlju 9.1 i opšte principe dizajna opisane u Poglavlju 1.4. Zbog toga neće biti detaljnije analizirane, već ih ukratko definišemo i ukazujemo na povezane heuristike i principe. Principi dizajna [10] su opštiji u smislu da se odnose i na vizuelni dizajn. Definicija heuristika upotrebljivosti [73] je takva da ih čini pogodnijim za testiranje interfejsa sa krajnjim korisnicima. Važno je napomenuti da su ih različiti istraživači definisali i predložili nezavisno jedni od drugih.

Osam zlatnih pravila dizajna korisničkog interfejsa glasi [79]:

- Konzistentnost (eng. *strive for consistency*) odgovara heuristici Konzistentnost i standardi (Poglavlje 9.1.4).

- Univerzalna upotrebljivost (eng. *seek universal usability*) govori da interfejs mora podržati korisnike sa različitim iskustvima korišćenja, slično heuristici Fleksibilnost i efikasnost korišćenja (Poglavlje 9.1.7).
- Povratna informacija (eng. *offer informative feedback*) je u skladu sa istoimenim principom dizajna (Poglavlje 1.4.2) i heuristikom Vidljivost statusa sistema (Poglavlje 9.1.1).
- Dijalog za zaključivanje (eng. *design dialogue to yield closure*) znači da interfejs mora biti intuitivan za korišćenje, odnosno da korisnik mora birati akcije na osnovu logike, a ne pogađanjem ili eliminacijom opcija. U tom podledu se oslanja na heuristike Usaglašenost sistema sa realnim svetom (Poglavlje 9.1.2) i Konzistentnost i standardi (Poglavlje 9.1.4).
- Prevencija grešaka (eng. *prevent errors*) se podudara sa istoimenom heuristikom upotrebljivosti (Poglavlje 9.1.5).
- Jednostavno poništavanje akcija (eng. *permit easy reversal of actions*) predstavlja element heuristike upotrebljivosti Kontrola i sloboda u korišćenju (Poglavlje 9.1.3).
- Doživljaj kontrole korisnika (eng. *keep users in control*) podrazumeva kontrolu nad interfejsom gde korisnik inicira akcije nad korisničkim interfejsom i dobija očekivane rezultate u pogledu logike akcija. Slaže se sa heuristikom upotrebljivosti Kontrola i sloboda u korišćenju (Poglavlje 9.1.3).
- Smanjeno opterećenje kratkoročne memorije (eng. *reduce short-term memory load*) je u skladu sa heuristikom upotrebljivosti Prepoznavanje pre pamćenja (Poglavlje 9.1.6).

Tabela 8 prikazuje podudaranje principa dizajna GKI (kolone) i heuristika upotrebljivosti (redovi).

Tabela 8. Uporedni prikaz i podudaranje principa dizajna GKI i heuristika upotrebljivosti.

	Konzistentnost	Univerzalna upotrebljivost	Povratna informacija	Dijalog za zaključivanje	Prevenција grešaka	Jednostavno poništavanje akcija	Doživljaj kontrole korisnika	Smanjeno opterećenje kratkoročne memorije
H1: Vidljivost statusa sistema								
H2: Usaglašenost sistema sa realnim svetom								
H3: Kontrola i sloboda u korišćenju								
H4: Konzistentnost i standardi								
H5: Prevenција grešaka								
H6: Prepoznavanje pre pamćenja								
H7: Fleksibilnost i efikasnost korišćenja								
H8: Estetski i minimalistički dizajn								
H9: Prepoznavanje i oporavak od grešaka								
H10: Pomoć i dokumentacija								

11. Literatura

1. Hewett et. al., (1996). ACM SIGCHI Curricula for Human-Computer Interaction, p. 5. Association for Computing Machinery Special Interest Group on Computer-Human Interaction.
2. Diaper, D. (1989). The discipline of HCI, *Interacting with Computers*, 1(1).
3. Bush, V., (1945) As We May Think, *Atlantic Monthly*.
4. Video prezentacija demonstracije NLS sistema (1968), The Mother of All Demos, presented by Douglas Engelbart. Dostupno: <https://cutt.ly/ed5dOux> (15.11.2020).
5. Obrenović, Ž. (2018). Quotes from IEEE Software History, *IEEE Software*, 35(5), pp. 10-13.
6. Jordan, P. W. (2002). *Designing pleasurable products: An introduction to the new human factors*. CRC press.
7. Nielsen, J. (1994). *Usability engineering*, Morgan Kaufmann.
8. Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology, *MIS quarterly*, pp. 319-340.
9. Deterding, S., et. al., (2011). Gamification. Using game-design elements in non-gaming contexts, In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pp. 2425-2428.
10. Norman, D., (2013). *The Design of Everyday Things: Revised and Expanded Edition*, MIT Press.
11. Google Material Design kolekcija ikona. Dostupno: <https://material.io/resources/icons/?style=baseline> (15.11.2020).
12. Sharp, H., Preece, J., and Rogers, Y. (2019). *Interaction Design: Beyond Human-Computer Interaction*, 5th Edition, Wiley Publishing.
13. Jovanović, M., Baez, M., and Casati, F. (2020), Chatbots as conversational healthcare services, *IEEE Internet Computing*.
14. McDuff, D., and Czerwinski, M. (2018). Designing emotionally sentient agents, *Communications of the ACM*, 61(12), pp. 74-83.
15. Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation, *Behaviour & Information Technology*, 1(3), pp. 237-256.
16. Harms, J. G., Kucherbaev, P., Bozzon, A., and Houben, G. J. (2019). Approaches for Dialog Management in Conversational Agents, *IEEE Internet Computing*, 23(2), pp. 13-22.
17. Petrović, J., and Jovanović, M. (2020). Conversational Agents for Learning Foreign Languages: a Survey, *Sinteza 2020 - International Scientific Conference on Information Technology and Data Related Research*, pp. 14 - 22.

18. Piwek, L., Ellis, D.A., Andrews, S., and Joinson, A. (2016). The Rise of Consumer Health Wearables: Promises and Barriers, *PLoS Med* 13(2): e1001953.
19. Baldi, T. L., Scheggi, S., Aggravi, M., and Prattichizzo, D. (2017). Haptic guidance in dynamic environments using optimal reciprocal collision avoidance, *IEEE Robotics and Automation Letters*, 3(1), pp. 265-272.
20. De Angeli, A., Jovanović, M., McNeill, A., and Coventry, L. (2020). Desires for active ageing technology. *International Journal of Human-Computer Studies*, 138, 102412.
21. Jovanović, M., Starčević, D., and Jovanović, Z. (2014). Reusable Design of Data Visualization Software Architecture for Unmanned Aerial Vehicles, *Journal of Aerospace Information Systems*, 11(6), pp. 359-371.
22. Parker, G. (2012). Acta is a four-letter word, *Acta Psychiatrica Scandinavica*, 126(6), pp. 476-478.
23. Reason, J. (1990), *Human Error*. Cambridge University Press, Cambridge.
24. Card, S. K., Moran, T. P., and Newell, A. (1983). *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates Inc.
25. Roth, I. (1986). *An introduction to object perception. Perception and Representation: A Cognitive Approach*. Red. I. Roth-JP Frisby, Milton Keynes, Open University Press.
26. Eyal, N., and Hoover, R. (2014). *Hooked: How to build habit-forming products*. Portfolio/Penguin.
27. Barry, T.E. (1987). The development of the hierarchy of effects: an historical perspective.
28. Shannon, C. (1948). *A Mathematical Theory of Communication*, Bell System Technical Journal.
29. Johnson, J. (2014). *Designing with the mind in mind: simple guide to understanding user interface design guidelines*, 2nd edition, Morgan Kaufmann Publishing.
30. Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., and von der Heydt, R. (2012). A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization, *Psychological bulletin*, 138(6), p. 1172.
31. Zvanični Veb sajt sa kolekcijom slika koje koriste princip figura/pozadina. Dostupno: <https://mcescher.com/gallery/symmetry/> (15.11.2020).
32. Larman, C., and Basili, V. R. (2003). Iterative and Incremental Developments. A Brief History, *IEEE Computer*, 36(6), pp. 47-56.
33. Royce, W. (1970). Managing the Development of Large Software Systems, *Proceedings of IEEE WESCON*, pp.1-9.
34. MockPlus alat za protipiziranje korisničkih interfejsa. Dostupno: <https://www.mockplus.com/> (15.11.2020).
35. Marvel platforma za protipiziranje korisničkih interfejsa. Dostupno: <https://marvelapp.com/> (15.11.2020).

36. Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*. John Wiley & Sons, New York, NY.
37. Angular okvir za razvoj Veb korisničkih interfejsa. Dostupno: <https://angular.io/> (15.11.2020).
38. Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., and Klemmer, S. R. (2010). Parallel prototyping leads to better design results, more divergence, and increased self-efficacy, *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(4), pp. 1-24.
39. Tohidi, M., Buxton, W., Baecker, R., and Sellen, A. (2006). Getting the Right Design and the Design Right, In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 1243-1252.
40. Benyon, D. (2014). *Designing interactive systems: A comprehensive guide to HCI, UX and interaction design*, Pearson Publishing.
41. Sutcliffe, A. (2012). *User-centred requirements engineering*. Springer Science & Business Media.
42. Uebersax, J. S. (2006). Likert scales: dispelling the confusion, *Statistical Methods for Rater Agreement*, 31.
43. Friberg, O., Martinussen, M., and Rosenvinge, J. H. (2006). Likert-based vs. semantic differential-based scorings of positive psychological constructs: A psychometric comparison of two versions of a scale measuring resilience, *Personality and Individual Differences*, 40(5), pp. 873-884.
44. Fontana A., and Frey J. (2005). The interview: from neutral stance to political involvement. In: N Denzin, Y Lincoln, eds. *The Sage Handbook of Qualitative Research*. 3rd edition, pp. 695– 727.
45. Gugl analitike alati: <https://analytics.google.com/> (15.11.2020).
46. Ivanov, I., i Jovanović, M. (2020). *Društvene mreže za kreativno angažovanje kroz zajedničke aktivnosti*, Diplomski rad, Univerzitet Singidunum.
47. Braun, V., and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), pp. 77-101.
48. Bergman, E. (2000). Designing the PalmPilot: a conversation with Rob Haitani. *Information appliances and beyond: interaction design for consumer products*, Morgan Kaufmann, pp. 81-102.
49. Cooper, A. (1994). The perils of prototyping. *Visual Basic Programmers Journal*.
50. Krug, S., (2014). *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (Voices That Matter)*, 3rd Edition, New Riders Publishing.
51. Hendriks-Jansen, H. (1996). *Catching ourselves in the act: Situated activity, interactive emergence, evolution, and human thought*, MIT Press.
52. Lim, Y. K., Stolterman, E., and Tenenberg, J. (2008). The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design

- ideas, ACM Transactions on Computer-Human Interaction (TOCHI), 15(2), pp. 1-27.
53. Doživljaj korisnika, Nielsen Norman Group: <https://www.nngroup.com/articles/definition-user-experience/> (15.11.2020).
 54. Ogonowski, C., et al., (2016). ICT-based fall prevention system for older adults: qualitative results from a long-term field study, ACM Transactions on Computer-Human Interaction (TOCHI), 23(5), pp. 1-33.
 55. Adair, J. G. (1984). The Hawthorne effect: a reconsideration of the methodological artifact, Journal of Applied Psychology, 69(2), p. 334.
 56. Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics, In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 152-158.
 57. Heuristička evaluacija, Nielsen Norman Group: <https://www.nngroup.com/topic/heuristic-evaluation/> (15.11.2020).
 58. MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction, Human-computer interaction, 7(1), pp. 91-139.
 59. Jenö, L. M., Vandvik, V., Eliassen, S., and Grytnes, J. A. (2019). Testing the novelty effect of an m-learning tool on internalization and achievement: A Self-Determination Theory approach, Computers & Education, 128, pp. 398-413.
 60. Dumas, J. S., and Redish, J. (1999). A practical guide to usability testing. Intellect books.
 61. Broj ispitanika u otkrivanju problema upotrebljivosti, Nielsen Norman Group: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (15.11.2020).
 62. Broj ispitanika u kvantitativnom testiranju upotrebljivosti, Nielsen Norman Group: <https://www.nngroup.com/articles/quantitative-studies-how-many-users/> (15.11.2020).
 63. Granične vrednosti u kvantitativnom testiranju upotrebljivosti, Nielsen Norman Group: <https://www.nngroup.com/articles/outliers-and-luck-in-user-performance/> (15.11.2020).
 64. Hartmann, J., Sutcliffe, A., and De Angeli, A. (2007). Investigating attractiveness in web user interfaces, In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 387-396.
 65. De Angeli, A., Sutcliffe, A., and Hartmann, J. (June). Interaction, usability and aesthetics: what influences users' preferences?, In Proceedings of the 6th conference on Designing Interactive systems, pp. 271-280.
 66. Gnanadesikan, R. (1997). Methods for Statistical Data Analysis of Multivariate Observations, 2nd Edition. Wiley Publishing.
 67. T-test definicija i varijante, Investopedia: <https://www.investopedia.com/terms/t/t-test.asp> (15.11.2020).

68. Jevremovic, A., Sarac, M., Milosavljevic, M., and Veinovic, M. (2014). Analyzing Behavior Of Students During The Electronic Testing, 1st International Conference on Electrical, Electronic and Computing Engineering.
69. Jevremović, A., Adamović, S., i Veinović, M. (2013). Praćenje kursora miša posetilaca kao evaluacija efikasnosti dizajna veb sajta, Zbornik radova 57. konferencije za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku - ETRAN.
70. Antonijevic, M., Shimic, G., Jevremovic, A., and Veinovic, M. (2018). Potentials of using artificial intelligence and EEG data in electronic assessments, 5th International Conference on Electrical, Electronic and Computing Engineering.
71. Jevremovic, A., Arsic, S., Antonijevic, M., Ioannou, A., and Garcia, N. (2018). Human-Computer Interaction Monitoring and Analytics Platform – Wisconsin Card Sorting Test Application, HealthyIoT 2018 - 5th EAI International Conference on IoT Technologies for HealthCare.
72. Modeli dizajna za specifična ponašanja korisnika (Laws of UX): <https://lawsofux.com/> (15.11.2020).
73. Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proceedings of ACM CHI'90 Conference, pp. 249–256.
74. Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. (2003). Heuristic evaluation of ambient displays, In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 169-176.
75. Pinelle, D., Wong, N., and Stach, T. (2008). Heuristic evaluation for games: usability principles for video game design, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1453-1462.
76. Sutcliffe, A., and Gault, B. (2004). Heuristic evaluation of virtual reality applications. Interacting with computers, 16(4), pp. 831-849.
77. Budd, A., (2007). Web Heuristics: http://www.andybudd.com/archives/2007/01/heuristics_for_modern_web_application_development/ (15.11.2020).
78. Nielsen, J., and Budiu, R. (2013). Mobile usability, New Riders Publishing.
79. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., and Elmqvist, N. (2016). Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th Edition, Pearson Publishing.
80. Kahn, P., and Lenk, K. (1998). Design: principles of typography for user interface design. Interactions, 5(6), p. 15.
81. Pravilo Trećina u dizajnu GKI, Interaction Design Foundation: <https://www.interaction-design.org/literature/article/the-rule-of-thirds-know-your-layout-sweet-spots> (15.11.2020).
82. F obrazac čitanja korisničkog interfejsa, Nielsen Norman Group:

- <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>
(15.11.2020).
83. Markus, N., i Jovanović, M., (2020). Društvene mreže za osobe sa posebnim potrebama, Diplomski rad, Univerzitet Singidunum.
 84. Bertin, J. (1981). Graphics and Graphics Information Processing, De Gruyter Publishing.
 85. Jennings, S. (2003). Artist's color manual: The complete guide to working with color. Chronicle Books.
 86. Basalla, G. (1988). The evolution of technology. Cambridge University Press.
 87. Seckler, M., Heinz, S., Bargas-Avila, J. A., Opwis, K., and Tuch, A. N. (2014). Designing usable web forms: empirical evaluation of web form improvement guidelines, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1275-1284.
 88. Nah, F. F. H. (2004). A study on tolerable waiting time: how long are web users willing to wait?, Behaviour & Information Technology, 23(3), pp. 153-163.

Spisak slika

Slika 1. Memex - nacrt personalnog računara za obavljanje kancelarijskih poslova. Preuzeto iz [3].	8
Slika 2. (a) Daglas Engelbart koristi NLS sistem; (b) uređaj miš koji je sistem koristio. Izvor: Michael Hicks from Saint Paul, MN, USA / CC BY.	9
Slika 3. Usporedni pregled razvoja sposobnosti čoveka i računara. Preuzeto iz [5].	10
Slika 4. (a) Demonstracija Skečped sistema; (b) Prva kaciga za virtuelnu stvarnost. Izvor: Dave Pape, CO, USA / CC BY.	11
Slika 5. (a) Xerox Star radna stanica; (b) Apple Macintosh radna stanica. Izvor: Dave Pape, CO, USA / CC BY.	11
Slika 6. Hijerarhija potreba korisnika tehnologije kao funkcionalnost (korisnost), upotrebljivost, i zadovoljstvo (doživljaj korisnika). Preuzeto i adaptirano iz [6].	14
Slika 7. Primer vidljivosti - Ribon (eng. <i>ribbon</i>) alata <i>Microsoft PowerPoint</i> predstavlja skup povezanih panela sa preglednim i jasno vidljivom opcijama za formatiranje teksta.	20
Slika 8. Povratna informacija - Klikom na dugme kreira se vizuelni efekat kružnog talasa (eng. <i>ripple effect</i>).	21
Slika 9. Primer ograničenja - Dostupne stavke menija za ažuriranje teksta alata <i>Microsoft PowerPoint</i> u trenutku kada nije selektovan tekst za obradu.	22
Slika 10. Primer ograničenja - Logičan raspored kontrola za slušanje muzike alata <i>iTunes</i> .	22
Slika 11. Primer metafore. Klizač fizičkog uređaja (a) predstavlja metaforu iz realnog sveta koja se koristi u dizajnu korisničkog interfejsa (b).	24
Slika 12. <i>Amazon Alexa</i> pametni zvučnici i mikrofoni za zadavanje glasovnih komandi virtuelnom asistentu. Izvor: Raysonho @ Open Grid Scheduler / Scalable Grid Engine CC0.	28
Slika 13. Primer vizuelnog komandnog interfejsa - <i>Adobe Illustrator</i> , alat za obradu slika, ima korisnički interfejs sastavljen od tematskih grupa alata kao ikona za instrukcije koje predstavljaju specifične operacije nad vizuelnim sadržajem.	29
Slika 14. Primer konverzacije - Woebot je virtuelni konverzacijski agent koji ima ulogu mentalnog terapeuta. Interakcija je zasnovana na tekstualnom dijalogu sa korisnikom (a) i podržana dodatnim funkcionalnostima (b).	30
Slika 15. Primer direktne manipulacije - <i>Apple</i> operativni sistem iz 1984. godine je koristio sisteme prozora i omogućavao prevlačenje objekata korisničkog interfejsa. Izvor: Andy F. Mesa / The Early Mac OS http://applemuseum.bott.org/sections/os.html (15.11.2020).	32
Slika 16. Virtuelna stvarnost je zasnovana na manipulaciji i navigaciji kao tipu interakcije. Izvor: European Space Agency, Reality Check, CC BY-SA 3.0 IGO.	33

Slika 17. Primer eksploracije i pretraživanja - <i>Amazon</i> sajt za kupovinu omogućava efikasnu pretragu proizvoda po različitim kriterijumima i njihov pregled.....	34
Slika 18. Dodir kao stil interakcije - Interaktivna tabla u učionici kao dodirni korisnički interfejs. Izvor: By Geofflambeth - Own work / CC BY-SA 4.0.	36
Slika 19. Primene i senzori za realizaciju oseta kao stila interakcije. Preuzeto iz [18].	38
Slika 20. Osetni korisnički interfejs za kreiranje muzičkih melodija kroz igru. Izvor: By Daniel Williams from NYC, USA - The Reactable / CC BY-SA 2.0.	39
Slika 21. Pomoć organizovana kao sistem pitanja i odgovora sa funkcijom pretrage gde su rezultati rangirani po učestalosti pitanja korisnika.	42
Slika 22. Primer pamćenja - <i>Excel</i> operatori za matematičke proračune zahtevaju učenje specifične sintakse, dok sa druge strane ubrzavaju rad korisnika.	44
Slika 23. Primer prepoznavanja - Meni kao element korisničkog interfejsa sadrži stavke opisane tekstualnim nazivima akcija (labelama). Labela predstavlja vizuelni trag koji opisuje akciju koju stavka pokreće.....	44
Slika 24. Modeli interfejsa i korisnika - Dva različita interfejsa koriste identičan simbol (horizontalne strelice) za dve različite operacije: (a) kretanje kroz istorijat posećenih stranica i (b) kretanje kroz skup rezultata pretrage rangiranih po relevantnosti.	46
Slika 25. Način grupisanja informacija u logičke jedinice koje čovek formira utiče na lakoću pamćenja informacija.	47
Slika 26. Jedinica informacije predstavlja opis pojedinačnog proizvoda koji kombinuje vizuelni sadržaj (fotografija i simboli) sa tekstualnim opisom karakteristika.	48
Slika 27. Primer prečica (a) tastature i (b) pokreta.	49
Slika 28. Podrazumevane vrednosti su inicijalno selektovane kako bi dodatno ubrzale korišćenje interfejsa.	50
Slika 29. Istorijat korišćenih objekata kao što su (a) dokumenti, (b) direktorijumi, ili (c) fontovi ubrzava korišćenje interfejsa.	51
Slika 30. Primer agregacije - (a) upravljanje porukama elektronske pošte i (b) prevlačenje grupe objekata korišćenjem miša.....	52
Slika 31. Tipologija i zastupljenost grešaka. Veličina oblika elipse Ven dijagrama odgovara udelu odgovarajuće vrste grešaka. Preuzeto i adaptirano iz [23].	54
Slika 32. Primer greške sličnosti akcija - tri ikone za označavanje poruke imaju različito značenje (eng. <i>check</i> , <i>start</i> , <i>important</i>), dele slične vizuelne karakteristike i raspoređene su blizu.	55
Slika 33. Balans brzine i kvaliteta korišćenja interfejsa je linearna funkcija. Brže korišćenje generalno povećava verovatnoću nastanka greške.	

Iskustvo čini brže korišćenje i preciznijim. Oblik krive zavisi i od konkretnog zadatka.	55
Slika 34. Dijalog potvrde koji podseća korisnika da sačuva promene pre zatvaranja dokumenta.....	56
Slika 35. Poruke o grešci: (a) poruka je konstruktivna tako što ukazuje korisniku na ispravan upit; (b) poruka sadrži istaknutu reč greška, implicira krivicu korisnika i sadrži tehničke termine (oznaka greške); i (c) poruka nije dovoljno precizna.	58
Slika 36. Model procesiranja čoveka. Preuzeto i adaptirano iz [24].....	61
Slika 37. Upravljanje pažnjom - (a) podeljena pažnja kod koje je fokus istovremeno i naizmenično na krivu i formule, i (b) približavanjem formula krivoj informacije se lakše i brže opažaju zbog toga što je fokus na jednoj celini (selektivna pažnja). Izvor: Computer Graphics course, Univerzitet Singidunum.	64
Slika 38. Određivanje maksimalnog informacionog potencijala korisničkog interfejsa korišćenjem Šenonove entropije.	68
Slika 39. Princip blizine. Na levoj strani se opažaju redovi, dok se na desnoj strani opažaju kolone. Preuzeto i adaptirano iz [29].	70
Slika 40. Princip blizine - Dijalog za formatiranje oblika alata <i>Microsoft PowerPoint</i> koristi okvire i horizontalne linije za razdvajanje kako bi komande bile opažane kao logičke celine koje obavljaju specifične funkcije. Logički povezane komande su bliže jedna drugoj tako da se opažaju kao celine. Na primer, raspored teksta, prilagođavanje teksta, podešavanja margina i dugmad za promene, na desnoj strani dijaloga. ...	71
Slika 41. Princip sličnosti - zajednička osobina na osnovu kojih se objekti opažaju kao grupa je boja unutrašnjosti oblika (eng. <i>fill</i>). Preuzeto i adaptirano iz [29].	72
Slika 42. Princip sličnosti u dizajnu dijaloga za podešavanje izgleda stranice.	73
Slika 43. Princip sličnosti u dizajnu simbola sastavljenih od jednostavnijih oblika.	73
Slika 44. Princip kontinuiteta - (levo) simbol koji se sastoji od delova koji nisu spojeni se opaža kao slovo H; (desno) logo poznatog brenda se opaža kao akronim kompanije iako se sastoji od nepovezanih pravolinijskih segmenata. Izvor: By Paul Rand - Captured from the front page of the IBM Notice of 2007 Annual Meeting and Proxy Statement.	74
Slika 45. Princip kontinuiteta - Klizač kao kontrola korisničkih interfejsa koja se koristi za definisanje vrednosti specifičnih parametara u podešavanjima.	75
Slika 46. Princip ograničavanja - nepovezani segmenti se opažaju kao celina.	76
Slika 47. Princip simetrije - objekat na slici opažamo kao prvu reprezentaciju na desnoj strani pre nego dve koje slede zbog toga što je jednostavnija. Preuzeto i adaptirano iz [29].	77

Slika 48. Princip simetrije kod opažanja trodimenzionalnih scena u dvodimenzionalnom prostoru. Izvor: Naslovna strana knjige Coherence in Thought and Action (Thagard, 2002).....	77
Slika 49. Princip figura/pozadina - kada se objekti preklapaju, potpuniji se opaža kao figura.	78
Slika 50. Princip figura/pozadina - sadržaj u pozadini Veb sajta može privući pažnju korisnika i zapravo predstavljati figuru. Izvor: Gorska služba spasavanja Srbije, http://gss.rs (15.11.2020).....	79
Slika 51. Princip figura/pozadina - modalne forme za ažuriranje podataka se mogu dodatno istaći tako što čine pozadinu aplikacije nedostupnom.	80
Slika 52. Princip pokreta - Aplikacije za vizuelizaciju podataka omogućavaju selekciju i prevlačenje grupe izborom proizvoljnog čvora klastera grafa. Čvorovi klastera se kreću u pravcu prevlačenja i opažaju kao grupa. Izvor: Disjoint Force-Directed Graph, D3js, https://observablehq.com/@d3/disjoint-force-directed-graph (15.11.2020).	81
Slika 53. Principi pokreta i sličnosti kod prevlačenja objekata na korisničkom interfejsu.....	82
Slika 54. Razvoj interaktivnog proizvoda se odvija u ciklusima (iteracijama).	85
Slika 55. Šematski prikaz Model Vodopada u razvoju softvera. Preuzeto i adaptirano iz Paul Smith's work at Wikipedia, CC BY 3.0.	86
Slika 56. Spiralni model razvoja korisničkog interfejsa - struktura i aktivnosti.	89
Slika 57. Spiralni model razvoja korisničkog interfejsa - (a) papirni (materijalni) prototip i, (b) digitalni (računarski) prototip.	90
Slika 58. Alternative dizajna korisničkog interfejsa za kontrolu klima uređaja - (a) cirkularni interfejs, (b) tabelarni interfejs, i (c) linearni interfejs. Preuzeto iz [39].	92
Slika 59. Definisanje zahteva korisnika u dizajnu interaktivnog sistema.	100
Slika 60. Funkcionalni zahtevi - UML dijagram slučajeva korišćenja (eng. <i>use case diagram</i>) se koristi za definisanje funkcija sistema.	100
Slika 61. Primeri odgovora upitnika zatvorenog tipa - (a) da/ne odgovor, i (b) odgovor sa skalom.....	104
Slika 62. Uporedni primer (a) likert i (b) semantičke diferencijalne skale.	105
Slika 63. Uporedni primer (1) sugerišućeg pitanja i (2) neutralno formulisano pitanja.	106
Slika 64. Dužina upitnika - (a) korisnici posvećuju više vremena pojedinačnim pitanjima kada je upitnik kraći; i (b) procenat nedovršenih upitnika se značajnije povećava za upitnike koji traju duže od 7-8 minuta.	108
Slika 65. Intervju - primeri pitanja (a) otvorenog i (b) zatvorenog tipa.	110
Slika 66. Tipičan tok intervjua sa aspekta angažovanja korisnika.	111
Slika 67. Primer Gugl Analitike alata za prikupljanje, analizu i vizuelizaciju podataka o aktivnostima korisnika bloga. Izvor: http://datahacker.rs/	112

Slika 68. Kvantitativna analiza - procenti su pogodni za ilustraciju udela i međusobnog odnosa podskupova podataka u okviru skupa (N=300 ispitanika). Preuzeto iz [46].	116
Slika 69. Kvantitativna analiza - srednje vrednosti (kolone) se mogu koristiti za kvantifikovanje osobina grupa podataka. Svaka grupa (red) opisuje deo korisničkog interfejsa Veb sajta.	117
Slika 70. <i>PalmPilot</i> uređaj (levo) i prototip od drveta (desno). Preuzeto i adaptirano iz [48].	119
Slika 71. 3D štampa se koristi za prototipiziranje (a) protetičkih uređaja, i (b) modela uređaja miša. Izvori: (a) https://www.seeedstudio.com/blog/ i (b) https://www.3dhubs.com/ (15.11.2020).	120
Slika 72. Prototipi u razvoju <i>SITU</i> aplikacije: (a) nacrtana skica sistema na papiru, (b) štampani vizuelni dizajn sistema, i (c) interaktivni digitalni prototip.	122
Slika 73. Prototipiziranje - (a) klasifikacija i (b) analiza prototipa na osnovu verodostojnosti kao složenog kriterijuma koji se može posmatrati po širini i dubini sistema koji se razvija. Preuzeto iz [49].	124
Slika 74. Papirni prototipi niske verodostojnosti se koriste na početku procesa razvoja - (a) skica interfejsa aplikacije društvenih mreža, i (b) nacrt interfejsa mobilne aplikacije za promovisanje fizičke aktivnosti.	126
Slika 75. Narativ koji ilustruje povezane elemente aplikacije kroz koje se korisnik može kretati.	127
Slika 76. Papirni prototipi su značajni za testiranje ideja i koncepata interfejsa u ranim fazama procesa razvoja.	128
Slika 77. Horizontalni prototip korisničkog interfejsa visoke verodostojnost kreiran korišćenjem alata za prototipiziranje <i>Mockplus</i> [34]. Prototip dostupan na: https://run.mockplus.com/SowmvpE1Ju701Ajk/index.html (15.11.2020).	130
Slika 78. Horizontalni prototip korisničkog interfejsa visoke verodostojnost kreiran korišćenjem Angular okvira za razvoj Veb korisničkih interfejsa [37].	132
Slika 79. Odnos broja ispitanika i identifikovanih problema upotrebljivosti sistema. 5 ispitanika može otkriti 80% ukupnog broja problema upotrebljivosti. Preuzeto iz [61].	146
Slika 80. Primeri zadataka za testiranje upotrebljivosti horizontalnog prototipa karpulng aplikacije za studente čiji su zahtevi opisani u Poglavlju 5.4. Zadaci su definisani sa aspekta ciljeva korisnika i testiraju: (a) funkciju registracije novog korisnika, (b) funkciju pretrage vožnji, (c) funkciju rezervacije vožnje, i (d) funkciju kreiranja vožnje.	147
Slika 81. Kvantitativno testiranje upotrebljivosti - odnos broja ispitanika i margine greške za specifične vrednosti pouzdanosti podataka. 20 korisnika može dati statistički pouzdane rezultate kvantitativnog testiranja upotrebljivosti. Preuzeto iz [61].	151

Slika 82. Prostorije za testiranje upotrebljivosti - (a) Laboratorija, odnosno eksperimentalno okruženje za testiranje upotrebljivosti. Izvor: https://interux.com/Usability-Lab-Estonia-Tallinn.html ; (b) Prostorija sa računarom. Izvor: https://www.testingtime.com/en/blog/nine-golden-rules-solid-user-testing/ (15.11.2020).	153
Slika 83. Tabelarni prikaz rezultata testiranja upotrebljivosti funkcija društvenih grupa korisničkog interfejsa. Kolone predstavljaju ozbiljnost kritične pojave, element interfejsa na koji se odnose, uspešna i neuspešna korišćenja elementa (greške), i sugestije za unapređenje (broj označava broj korisnika koji su dali komentar).	154
Slika 84. Bar dijagram koji daje uporedni prikaz srednje vrednosti subjektivnog estetskog doživljaja korisničkog interfejsa za različite ciljne grupe korisnika. Preuzeto iz [64].	154
Slika 85. Pita dijagram koji prikazuje odnos vrednosti odgovora upitnika u vezi sa korišćenjem aplikacija društvenih mreža (N=300 ispitanika). Preuzeto iz [46].	155
Slika 86. Linijski dijagram pogoduje praćenju trendova posmatranih vrednosti. Dijagram prikazuje uporedni prikaz kretanja dve vrednosti nezavisne varijable (kao specifični tipovi interfejsa) u odnosu na mere upotrebljivosti kao zavisne varijable. Preuzeto iz [65].	156
Slika 87. Način korišćenja stonog uređaja za praćenje pogleda <i>The Eye Tribe</i>	162
Slika 88. Primer prenosivog uređaja za snimanje pogleda <i>Tobii Pro Glasses 3</i> . Naočare (levo) poseduju kamere koje snimaju oči korisnika, kao i kameru koja snima ono što korisnik vidi. Naočare se povezuju sa uređajem (desno) koji snima podatke i obezbeđuje napajanje električnom energijom iz baterije.	163
Slika 89. Mobilni uređaj za praćenje pogleda korišćen u dokumentarnom filmu <i>Ronaldo: Tested to the Limit</i> (2011).	164
Slika 90. <i>WebGazer.js</i> biblioteka nudi mogućnost korišćenja Veb kamere za praćenje pogleda korisnika koji pristupaju Veb stranici.	165
Slika 91. Primer toplotne karte (eng. <i>heatmap</i>) dobijene praćenjem kursora miša posetilaca sajta racunarskemreze.com	166
Slika 92. Toplotne karte pogleda korisnika (levo) i kursora miša (desno) za istu sesiju (srednje/nisko-interaktivni sadržaj).	166
Slika 93. Viša korelacija između toplotnih karata pogleda korisnika (levo) i kursora miša (desno) u slučaju visoko-interaktivne aplikacije.	167
Slika 94. Primer vizuelne reprezentacije klikova.	167
Slika 95. Primer korišćenja <i>EEG</i> i <i>eye tracker</i> uređaja u eksperimentima vezanim za interakciju korisnika i računara.	168
Slika 96. Primer dijagrama izvedenih karakteristika mentalnog stanja korisnika na osnovu podataka dobijenih putem <i>EEG</i> senzora u eksperimentu trajanja 30 sekundi sa učestalošću uzorkovanja od 2Hz.	169

Slika 97. Arhitektura HCI.MAP platforme za izvođenje eksperimenata sa krajnjim korisnicima.	170
Slika 98. Kontrolni panel HCI.MAP platforme sa prikazom sesija, senzora i izvozom podataka.....	171
Slika 99. Da li je efikasnije postaviti glavni meni sa leve (A) ili sa desne strane (B) ? A/B testiranje može statistički utvrditi koja opcija je efikasnija za korisnike.....	172
Slika 100. Vidljivost statusa sistema - (a) Promena izgleda dugmeta prilikom klika; (b) Promena boje i dodavanje simbola () korisniku poručuje da je sistem registrovao izbore korisnika (levo). Indikator progressa govori korisniku da operacija sistema zahteva određeno vreme (desno).	178
Slika 101. Usaglašenost sistema sa realnim svetom - Jezik korisničkog interfejsa mora biti razumljiv krajnjim korisnicima.....	180
Slika 102. Usaglašenost sistema sa realnim svetom - Fizički kompas (desno) kao metafora za virtuelni kompas (levo).....	181
Slika 103. Usaglašenost sistema sa realnim svetom - Metafore se mogu naći i u standardnim elementima GKI kao što je kartica.	182
Slika 104. Usaglašenost sistema sa realnim svetom - Metafore mogu odrediti i celokupan doživljaj GKI.	182
Slika 105. Kontrola i sloboda u korišćenju - Aplikacija Gugl dokumenti (eng. <i>Google docs</i>) prati istorijat rada korisnika i daje hronološki spisak izmena kako bi se sadržaj mogao menjati od trenutka prošlog vremena.....	183
Slika 106. Konzistentnost i standardi - <i>Spotify</i> aplikacija za slušanje muzike pruža doživljaj konzistentnog izgleda i ponašanja prilikom korišćenja različitih prikaza interfejsa.	184
Slika 107. Konzistentnost i standardi - Praktični standardi izgleda i ponašanja elemenata korisničkog interfejsa koje propisuju okviri za razvoj doprinose konzistentnosti realizovanih aplikacija.	185
Slika 108. Prepoznavanje pre pamćenja - videti i izabrati (desno) je lakše nego pamtit i kucati (levo).	186
Slika 109. Prepoznavanje pre pamćenja - pravilno dizajnirani vizuelni simboli akcija i sadržaja interfejsa pospešuju prepoznavanje (a i b). Pored toga simboli se mogu izvesti iz sadržaja kako bi korisnik imao bolju sliku o celini bez pretraživanja, minijaturene slike (eng. <i>thumbnail</i>) pod (c).	187
Slika 110. Fleksibilnost i efikasnost korišćenja - forma za pretragu u desnoj vertikalnoj traci interfejsa grupiše kriterijume pretrage po kategorijama podataka. Izvor: <i>Unity Assets Store</i>	189
Slika 111. Fleksibilnost i efikasnost korišćenja - Dugmad u zaglavlju interfejsa su grupisana na osnovu funkcija koje obavljaju kao raspored interfejsa, dodavanje/uklanjanje particija, upravljanje particijom, i informacije o izabranoj particiji.....	190
Slika 112. Fleksibilnost i efikasnost korišćenja - simetrija kao balans sadržaja i ravnanje elemenata korisničkog interfejsa.	191

Slika 113. Pravilo Trećina definiše fokusne tačke ekrana u čijim okolinama se mogu smestiti specifični sadržaji interfejsa u zavisnosti od njihove važnosti. Preuzeto iz [81].	192
Slika 114. Pravilo Trećina se koristi u dizajnu korisničkog interfejsa za raspoređivanje elemenata sadržaja.	193
Slika 115. F obrazac skeniranja sadržaja GKI opisuje kombinacije vertikalnih i horizontalnih pokreta pogleda korisnika prilikom čitanja sadržaja interfejsa.	194
Slika 116. Estetski i minimalistički dizajn - Jednostavnost kao redukovanje elemenata korisničkog interfejsa aplikacije za slušanje muzike.	195
Slika 117. Regularnost u estetskom i minimalističkom dizajnu - Ponavljajući elementi se mogu istaći pozicijom i veličinom iz skupa (a). Ponavljajući šablon koji kombinuje elemente interfejsa opisuje omiljene aktivnosti korisnika u donjem delu interfejsa na desnoj strani (b). Ponavljajuće forme ikona sa labelama u središnjem delu (c). Preuzeto iz [83].	197
Slika 118. Estetski i minimalistički dizajn - podnožje tabele predstavlja element koji ima veći broj funkcija, odnosno uloga.	198
Slika 119. Estetski i minimalistički dizajn - (a) kontrast pomoću pozicije i veličine, i (b) kontrast pomoću pozicije, veličine, boje i oblika. Preuzeto iz [50].	199
Slika 120. Točak boja za realizaciju vizuelnog kontrasta - (a) kombinovanje osnovnih boja, i (b) kombinovanje nijansi osnovnih boja.	200
Slika 121. Struktura i oblasti čovekovog vidnog polja sa predlogom optimalne pozicije i veličine displeja.	205
Slika 122. Video igra <i>Super Mario Bros</i> iz 1985. godine kreirana je u rezoluciji 256x224 piksela i koristi 56 boja (standardno za <i>Nintendo Entertainment System</i> konzolu). Ukupna veličina igre je 31 KB.	207
Slika 123. Iscrtavanje linije na diskretnom displeju: vektorska instrukcija (levo), sa alijasingom (centar) i sa anti-alijasingom (desno).	208
Slika 124. Oštar prikaz rasterskog GNU unifont-a u predviđenoj veličini: 8x16 piksela (levo). Promena veličine fonta decimalnim množiocem smanjuje oštrinu i uvodi tonove sive boje (desno).	209
Slika 125. Prikaz teksta vektorskim fontom u visokoj rezoluciji bez anti-alijasinga (gore) i sa anti-alijasingom (dole).	209
Slika 126. Prikaz teksta vektorskim fontom u niskoj rezoluciji bez anti-alijasinga (gore) i sa anti-alijasingom (dole).	210
Slika 127. Izgled VGA, HDMI i DisplayPort konektora.	212
Slika 128. Primer “cepanja ekrana”, odnosno kombinovanog prikazivanja dva frejma zbog nedostatka vremenske sinhronizacije.	213
Slika 129. Različiti tipovi elemenata za unos teksta.	216
Slika 130. Primer ulaznog tekstualnog elementa za unos datuma i vremena pomoću kalendara.	216

Slika 131. Primer korišćenja WYSIWYG editora kao dopune kontrole za unos teksta u više redova.	217
Slika 132. Primeri elemenata GKI za izbor većeg broja ponuđenih opcija. ...	217
Slika 133. Dugme za čekiranje služi za podešavanja kod kojih opcije nisu međusobno isključive.	218
Slika 134. Naizmenično dugme (levo) je primer metafore iz realnog sveta (desno).	218
Slika 135. Radio dugme služi za izbor opcija koje međusobno isključive (a) i vodi poreklo iz realnog sveta (b).	219
Slika 136. Padajuća lista daje veći broj opcija od kojih se jedna može izabrati u trenutku. Naziv liste mora biti vidljiv i kada je lista prikazana.	221
Slika 137. Padajuća lista mora imati isti sadržaj za različite kontekste aplikacije. Opcije nedostupne za dati kontekst se mogu učiniti neaktivnim. U slučaju većeg broja opcija lista može biti kaskadna.	221
Slika 138. Liste podataka kombinuju slobodan unos i ponuđene stavke.	222
Slika 139. Primer korišćenja "mrvice hleba" kao elementa navigacije u Veb aplikaciji.	223
Slika 140. Kartica je navigacioni element sekundarnog nivoa koji grupiše i prikazuje sadržaje u okviru jednog konteksta ili celine.	224
Slika 141. Meni grupiše i prikazuje akcije na zahtev. Redosled akcija obično određuje relevantnost u pogledu učestalosti pozivanja. U slučaju većeg broja mogućih akcija mogu se koristiti kaskadni meniji.	225
Slika 142. Standardan izgled simbola ikone za oznaku "hamburger" menija.	225
Slika 143. Straničenje podataka u tabeli, sa mogućnošću izbora broja rezultata po stranici, informacijom o trenutnoj stranici, mogućnošću skoka na određenu stranicu i navigacijom na prvu/prethodnu/sledeću/poslednju stranicu.	226
Slika 144. Straničenje se koristi za prikaz rezultata pretrage ukoliko se oni ne mogu prikazati na jednoj strani interfejsa.	227
Slika 145. Primer korišćenja pretrage za izdvajanje (filtriranje) željenih informacija.	227
Slika 146. Varijante klizača kao (a) kontinualan i (b) diskretan.	228
Slika 147. Određeni i neodređeni progres elementi GKI.	229
Slika 148. Jedan od prvih računara koji su imali GKI sa prozorima, <i>Xerox Alto</i> , razvijen je početkom 70-ih godina 20. veka.	231
Slika 149. <i>Gimp</i> , softverski alat otvorenog koda za obradu rasterske grafike, ima mogućnost da sve elemente prikazuje unutar jednog prozora (levo) ili da glavne elemente i otvorene datoteke prikazuje u odvojenim prozorima (desno).	232
Slika 150. Primer grupisanja prozora iste aplikacije na liniji zadataka u <i>XFCE</i> grafičkom korisničkom okruženju.	232
Slika 151. Primer korišćenja nemodalnog prozora, odnosno prozora koji ne blokira ostali sadržaj.	233

Slika 152. Iskačući dijalozi za obaveštavanje, dobijanje potvrde i unos vrednosti od strane korisnika, podrazumevano podržani u svim Veb pretraživačima koji imaju podršku za JavaScript.	233
Slika 153. Primer upotrebe modalnog dijaloga umesto preusmeravanja korisnika na novu stranicu sa formom.	234

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

004.5(075.8)

ЈОВАНОВИЋ, Млађан, 1981-

Interakcija čovek - računar / Mlađan Jovanović, Aleksandar Jevremović.
- 1. izd. - Beograd : Univerzitet Singidunum, 2020 (Beograd : Caligraph). -
253 str. : ilustr. ; 24 cm

Na vrhu nasl. str.: Fakultet za informatiku i računarstvo, Tehnički
fakultet. - Tiraž 300. - Bibliografija: str. 238-243. - Registar.

ISBN 978-86-7912-738-9

1. Јевремовић, Александар, 1983- [аутор]

а) Интеракција човек-рачунар

COBISS.SR-ID 26450697

© 2020

Sva prava zadržana. Nijedan deo ove publikacije ne može biti reprodukovan u bilo kom vidu i putem bilo kog medija, u delovima ili celini bez prethodne pismene saglasnosti izdavača.



Mladen Jovanović
Aleksandar Jevremović

INTERAKCIJA ČOVEK-RAČUNAR

Razvojem i unapređenjem tehnologije, tema interakcije čoveka i računara je aktuelna i sveprisutna u različitim oblastima informatike i naukama poput psihologije, medicine, jezika, sociologije i drugih. Pored toga, oblast prožima različite aspekte naših života, privatne i poslovne. Možemo reći da osim što mi kao ljudi oblikujemo tehnologiju, tehnologija zauzvrat oblikuje nas i naše živote. Zbog ove uzajamne relacije oblast ima značajno mesto u savremenom društvu.

Udžbenik najpre daje širi kontekst oblasti kroz njen razvoj i postulate. Zatim opisuje načine na koje koristimo tehnologiju u svakodnevnom životu. Nakon toga opisujemo principe korišćenja tehnologije sa aspekta cilja koji je zajednički svima nama, a to je da je to što koristimo lako i prijatno za upotrebu. Razumevanju principa doprinosi poznavanje osnovnih psiholoških mehanizama korišćenja tehnologije. Sledi opis razvoja interaktivnih tehnologija koji poštuje prethodno opisane principe. Najpre predstavljamo proces razvoja kao celinu, a zatim detaljnije obrađujemo njegove delove kao što su definisanje korisničkih zahteva, protopiziranje i testiranje interaktivnih tehnologija. Na kraju analiziramo korisničke i tehničke detalje grafičkih korisničkih interfejsa putem kojih najčešće upotrebljavamo tehnologiju.