

**Keywords:** human-computer interaction, user-interface, user-interface style, direct manipulation, user-interface design

Matjaž Debevec, Dali Đonlagić, Martin Leš  
Tehniška fakulteta Maribor  
Elektrotehnika, računalništvo in informatika

## POVZETEK

Komunikacija človek-računalnik postaja vse pomembnejša na vedno več področjih. Pomemben del te komunikacije je uporabniški vmesnik, ki omogoča prenos informacij med človekom in računalnikom. Uspešnost uporabniško prijaznega programa je odvisna predvsem od preprostosti in hitrosti njegove uporabe. Z razvojem računalništva so nastajali različni slogi uporabniških vmesnikov, v začetku alfanumeričnih, nato grafičnih.

Uspešnost uporabniško prijaznih programov je v veliki meri odvisna od pravilne izbire elementov iz različnih slogov uporabniških vmesnikov. V članku so podani in opisani najpogostejši slogi uporabniških vmesnikov. Opisane so njihove lastnosti in ideje, njihove prednosti in slabosti in povezave med različnimi slogi.

## USER-INTERFACE STYLES

Human-Computer communications is becoming important in many fields. Important part of this communication is user interface, which enables the information transmission between man and the computer. The efficiency of the user friendly program depends on simplicity and speed of its use. With the development of software engineering various user interface styles, first the alpha-numerical then the graphical one, have appeared.

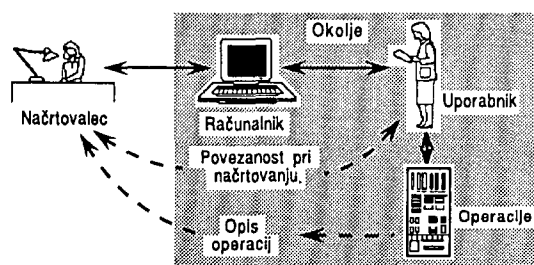
The efficiency of user-friendly programmes depends on elements, correctly chosen from various user interface styles. In the paper the most frequently used user interface styles are given and described. Their characteristics and ideas, their advantages and weaknesses and connections between various styles are described.

## 1. Uvod

Začetki komunikacije človek-računalnik segajo v 60 leta, ko so poskušali razširiti uporabo računalnikov na druga področja. S tem se je močno razširil krog uporabnikov. Leta 1963 so začeli razvijati prve sisteme z dodeljevanjem časa, od katerih je najbolj znan MIT-MAC. Od takrat se je počasi povečal pomen ostalih tipov uporabnikov, ki so želeli uporabljati računalnike. Temu ustrezno so se razvijali različni slogi uporabniških vmesnikov. Najopaznejši premik v razvoju uporabniških vmesnikov je nastal z uvajanjem grafičnih delovnih postaj v 80 letih<sup>1</sup>.

Uporabniški vmesnik je splošno gledano informacijski kanal, ki prenaša informacije med človekom in računalnikom. Vendar pri tem ne gre samo za celoten potek od razvoja programa do njegove uporabe in po potrebi do ponovnega načrtovanja za izpolnitev programa. Slika 1 kaže realen pretok informacij med načrtovalcem, računalnikom in uporabnikom. Načrtovalec načrtuje program tako, da prilagodi njegovo delovanje z uporabnikovim delom in zahtevami. Pri tem mora načrtovalec upoštevati tudi različne tipe uporabnikov. Nujna je začetna povezava načrtovalca z uporabnikom, saj ni nujno, da načrtovalec pozna vse zakonitosti uporabnikovega dela. Tudi po daljšem delu s programom se lahko

zgodí, da pride do sprememb, ki jih mora načrtovalec pozneje vstaviti v program. Vmesnik mora biti zato čim bolj odprt, kar pomeni, da bi lahko uporabljali enak vmesnik, na primer, za vodenje krmilnih naprav (industrijski procesi) ali za vodenje simuliranih naprav (simulator motorja). Zaradi tega ni enostavno izbrati ustreznega sloga uporabniškega vmesnika. V večini primerov izberemo, glede na okolje, v katerem bo uporabnik delal, elemente iz različnih slogov uporabniških vmesnikov. Postavlja se vprašanje, kateri slogi so najbolj primerni za določeno okolje. Zato mora načrtovalec dobro poznati njihove lastnosti, prednosti in slabosti. V nadaljevanju so opisani različni slogi uporabniških vmesnikov in njihove značilnosti<sup>2</sup>.



Slika 1: Pretok informacij

## 2. Slogi uporabniških vmesnikov

V zadnjem času nastopajo naslednji slogi uporabniških vmesnikov<sup>3</sup>:

- **WYSIWYG** ("What you see is what you get")- vidiš to, kar si podal,
- **direktna manipulacija**,
- **ikone**,
- **multimedialni dialog**.

Skupna, glavna značilnost zgornjih uporabniških vmesnikov je potek komunikacije v grafičnem načinu in je zato pri delu potrebna zmogljiva grafična postaja.

V tem poglavju opisujemo lastnosti in ideje posameznih slogov, njihove prednosti, slabosti in povezave med različnimi slogi. Na koncu je tudi kratek opis drugih slogov uporabniškega vmesnika, ki ne spadajo direktno pod grafične uporabniške vmesnike. Ti so:

- **Izbira po meniju**,
- **ukazna vrstica**,
- **dialog v naravnem jeziku**,
- **dialog s vprašanji in odgovori**.

Za uporabniške vmesnike lahko seveda uporabimo kombinacijo različnih slogov, vendar mora biti kombinacija pravilno izbrana in primerna glede na tip uporabnikov.

### 2.1. WYSIWYG slog

WYSIWYG (izgovori se "wiz-ee-wig") je zelo pomemben slog grafične predstavitve informacij. Predstavitve sistema, procesa, operacij, ki jih uporabnik vidi na zaslonu odgovarjajo realnemu, naravnemu izgledu. Večina dobrih programov, ki vsebujejo grafično predstavitve, vključujejo tudi komponente WYSIWYG.

Dobri urejevalniki besedil, na primer, uporabljajo WYSIWYG slog: Besedilo, ki ga želimo poudarjeno izpisati na tiskalnik, se tudi poudarjeno izpiše na zaslonu. Z urejevalnikom besedila, ki ne vsebuje WYSIWYG sloga, mora uporabnik uporabljati krmilne ukaze za besedilo.

Za primer si izpišimo naslednji stavek:

V tem stavku vidimo **<B>poudarjen<N>**, **<I>poševen<N>** in **<U>podčrtan<N>** tekst.

ki definira naslednji izpis na tiskalnik:

V tem stavku vidimo **poudarjen**, *poševen* in podčrtan tekst.

V tem primeru vidimo, da se moramo najprej naučiti pomena krmilnih ukazov in si ustvariti abstrakcijo izgleda na tiskalniku. Dokler uporabnik besedila ne izpiše, nima predstave o tem, ali je vse krmilne ukaze pravilno vpisal. Pri zelo dolgem besedilu je v takšnih primerih možnost napake zelo velika. Še največ krmilnih ukazov je potrebno pri matematičnih izrazih, kjer lahko nastopa nepregledna množica krmilnih kod. WYSIWYG slog v teh primerih bistveno izboljša pregled nad vnosom.

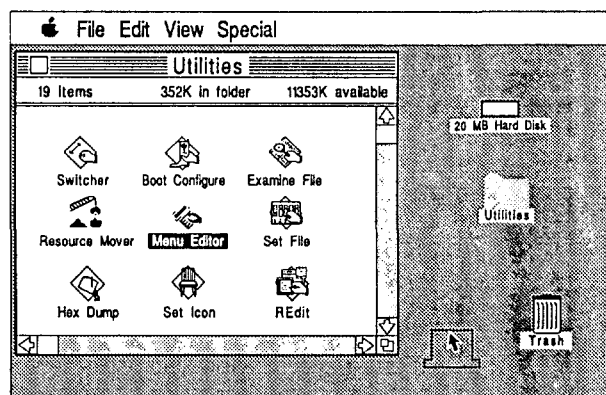
WYSIWYG slog ima tudi nekatere slabosti. Kjerkoli nastopajo zmogljive grafične postaje z veliko resolucijo in z mnogimi barvami, lahko nastopi problem, kako vse te informacije prenesti na papir, še posebej, če ni na voljo enako zmogljivega tiskalnika.

Lahko se pojavi tudi nasprotna situacija. Uporabnik ima na voljo zelo slab uporabniški vmesnik z WYSIWYG slogom, s katerim ne more točno vedeti, kaj se bo izpisalo na tiskalniku. Na voljo ima, na primer, v urejevalniku besedila, neko končno množico fontov, ki se prikažejo na zaslonu. Na tiskalniku pa ima možnost izpisa večje množice fontov, kot jih ima na zaslonu. Velikost in tip fonta na zaslonu zato včasih ne sovpadata s fontom na tiskalniku.

### 2.2. Direktna manipulacija

Uporabniški vmesnik direktne manipulacije je sestavljen iz **objektov**, njihovih lastnosti in iz medsebojnih povezav, nad katerimi izvajamo **operacije**. Te operacije so tudi vizuelno predstavljene in jih izvajamo s pomočjo vizuelnih interaktivnih tehnik, kot na primer s puščico na zaslonu, ki predstavlja pomik s pomočjo miške ali grafične table. Ukazov ne izvajamo v smislu izbire po meniju ali s tipkanjem ukazov, ampak jih izvajamo kot akcije vizuelne predstavitve. Ta predstavitve je lahko besedilo, kot na primer ime objekta ali lastnosti in **grafični simbol**, ki ga imenujemo tudi **ikona**. V nadaljevanju je podrobneje opisan pomen grafičnih simbolov, ikon in vizuelni predstavitvi.

Primer uporabniškega vmesnika direktne manipulacije je predstavljen na sliki 2. V zgornjem desnem kotu je ikona za trdi disk, pod njim je ikona direktorija, ki je v sivi barvi, kar pomeni, da je odprta. Na levi strani zaslona je prikaz odprtega direktorija s poimenovanimi ikonami, ki predstavljajo datoteke v tem direktoriju. Datoteka, od katere vidimo le obrise okoli puščice, se pomika proti ikoni za smeti, ki je v spodnjem desnem kotu.



Slika 2: Zaslon računalnika Macintosh

Disketne enote in datoteke so torej v tem slogu predstavljene kot ikone. Pomik ene ali več datotek iz ikone, ki predstavlja prvo disketno enoto v ikono, ki predstavlja drugo disketno enoto, je enak pomenu kopiranja teh datotek iz ene v drugo disketno enoto. Če, na primer, datoteko pomaknemo v ikono za smeti (Trash), se datoteka zbrše.

Izraz direktna manipulacija je prvi skoval Shneiderman<sup>4</sup>, ki je predstavil tudi mnogo različnih primerov, ki kažejo tipične značilnosti direktne manipulacije. Te so med drugim vnaprejšnja predstavitve besedila, oblike, kot je pozneje izpisan na tiskalniku (WYSIWYG), pri izpisu tabel so potrebne oblike, ki kratko in jasno predstavljajo podane tabele in drugo. Video igre, CAD/CAM sistemi, velike podatkovne baze, procesni sistemi so prav tako primeri programov, kjer najdemo direktno manipulacijo.

Shneiderman je predstavil tudi prednosti direktnih manipulacijskih tehnik:

- začetniki se hitreje naučijo osnovnih ukazov,
- izkušen uporabnik se lažje takoj posveti svojemu problemu,
- število sporočil o napakah se bistveno zmanjša,
- večja prilagodljivost različnim tipom uporabnikov.

Uporabniški vmesnik direktne manipulacije je sestavljen iz različnih komponent. Na najvišjem nivoju direktnega manipulacijskega sistema je grafična predstavitev **metafore**, kot na primer površina **pisalne mize**, ki predstavlja liste in orodja. Druge metafore so lahko predstavitev **modela procesnega sistema**, **elektronskega laboratorija**, **merilnega sistema** in drugih. Na tem najvišjem nivoju lahko uporabnik uporablja objekte, ki so prikazani v čim realnejši obliki v oknih na zaslonu. Oken je lahko na zaslonu poljubno mnogo in se lahko med seboj pokrivajo, lahko se priključijo in uporabljajo, tako kot bi ravnali s kupi listov na pisalni mizi.

Za manipulacijo objektov in vodenje uporabljamo različne načine, kot so:

- **upravljanje**,
- **vodenje**,
- **menuji**,
- **dialogna področja**.

**Upravljanje** pomeni direktno vodenje objekta, kot na primer merilnega instrumenta v programu za merjenje. To vodenje se lahko po izbiri ustreznega objekta z miško direktno opazuje na zaslonu. Izvršujemo lahko množico operacij, manipulacij, kot na primer, spremembo velikosti, razteg, rotacijo in drugo.

**Vodenje** pomeni izbira funkcij ali vnos različnih parametrov vhoda. Na voljo imamo različne oblike vodenja:

- **tipke** (navadne tipke, radio tipke, kontrolne tipke),
- **okvirji** (izpisni, vnosni),
- **valuatorji** (skale, drsni rob (Scroll bars), prirejeni valuatorji, primerni za določen program).

**Menuje** prištevamo med najelementarnejše dele vodenja, ki jih uporabljamo tudi v drugih slogih. Menuji lahko nastopajo tudi sami zase. V nekaterih uporabniških okoljih so menuji samo zbirka tipk, ki jih aktiviramo s pritiskom na miškino tipko. Najbolj pogosto uporabljene oblike menujev so **pull-down**, **pop-up** in **kaskadni menu**.

**Dialogna področja** uporabljamo za izpis sporočil za preverjanje ali potrditev pomembnih ukazov, kot na primer "Ali naj shranim datoteko pred izhodom iz programa?". Ta področja uporabimo tudi za vnos podatkov ali parametrov. Značilna lastnost teh področij je, da uporabnik najprej odgovori na vprašanje ali vstavi podatek, ki ga sporočilo zahteva, preden lahko dela dalje s programom. Ta tehnika je manj direktno manipulativna kot ostale, ker nudi nižji nivo komunikacije s človekom.

Ostali elementi direktne manipulacije so povezani s predstavitvijo informacije, kot na primer **ikone**, s katerimi lahko manipuliramo in izvajamo določene operacije.

Tipične **operacije** po izbiri objekta s pomočjo vhodne, izbirne naprave so:

- premik objekta na drugo lokacijo,
- brisanje objekta,
- kopiranje,
- rezanje in dodajanje odrezanega objekta,
- odpiranje in zapiranje datotek.

Direktno manipulacijo velikokrat predstavljajo kot najboljši slog uporabniškega vmesnika. Ta slog je v resnici izredno zmogljiv in zelo enostaven za učenje. Vendar lahko postane za izkušenega uporabnika počasen. Pri tiskanju datoteke s poljubnim imenom, na primer, mora uporabnik pri direktni manipulaciji najprej odpreti okno za izbiro datotek, poiskati datoteko in jo določiti z vhodno napravo (običajno z miško). Nato mora poiskati ikono ali izbrati ukaz za tiskanje. Če pa uporabnik že pozna ime datoteke in ve v katerem direktoriju se nahaja, je zanj mnogo hitreje, če vtipka ukaz: "Tiskaj datoteke". To je še posebej primerno pri velikem številu datotek, kjer bi se moral uporabnik pri iskanju datoteke v oknu direktorija pomikati po straneh. To je samo eden od mnogih primerov, pri katerih je vnos v ukazno vrstico lahko mnogo hitrejši kot direktna manipulacija. Omejitve se kažejo tudi v tem, da potrebuje direktna manipulacija zelo zmogljivo grafično postajo, skupaj z miško. Poleg tega ni enostavna realizacija slike, objektov in težjih ukazov na zaslonu, ki predstavljajo metaforo uporabnikovega delovnega okolja.

Priporočljivo je, da kombiniramo direktno manipulacijo z ukazno vrstico. V nekaterih primerih se celo zgodi, da nekaterih ukazov ne moremo vključiti v direktno manipulacijo. Zato mora uporabnik imeti možnost vpisa teh ukazov v ukazno vrstico. To pomeni, da se pri načrtovanju uporabniškega vmesnika ne opredelimo samo za en slog.

### 2.3. Uporabniški vmesnik z ikonami

**Ikona** je **slikovna predstavitev** objekta, akcije, lastnosti in drugih konceptov. Načrtovalec uporabniškega vmesnika ima za predstavitev določenega koncepta na voljo uporabo ikone ali besed. V tem primeru ikone ne smemo povezovati z direktno manipulacijo. Besedilo sicer lahko direktno manipuliramo enako dobro kot ikone, vendar ikone zmanjšajo potreben prostor za izpis.

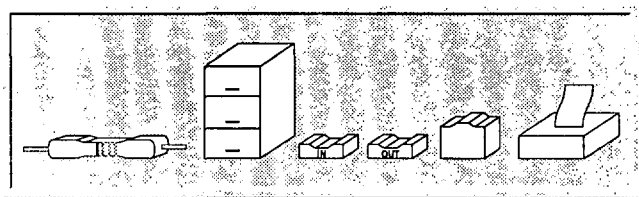
Vprašanje je, kaj je bolje uporabiti, ikone ali besedilo. Kot velika večina odgovorov, je tudi tukaj odgovor odvisen od same naloge in ciljev uporabnikovega dela. Ikone imajo veliko prednosti pred izpisom besedila. Dobro načrtovana ikona lahko pove mnogo več kot besede, zavzame mnogo manj prostora in je s tem razbiranje mnogo lažje. Če je ikona pravilno izbrana, lahko postane tudi jezikovno neodvisna, kar omogoča uporabo uporabniškega vmesnika v različnih državah.

Zato je potrebno pri načrtovanju upoštevati tri cilje:

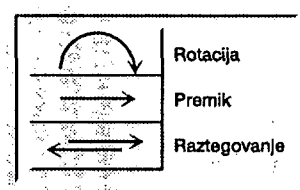
- **razpoznavanje** - pomeni hitrost razpoznavne pomena ikone,
- **pomnjenje** - pomeni, kako dobro si lahko uporabnik zapomni pomen ikone,
- **razlikovanje** - pomeni, kako dobro se ikone med seboj ločijo po izgledu in pomenu.

Ikone, ki predstavljajo objekte, se lahko relativno enostavno načrtujejo in jih imenujemo **objektne ikone**. Slika 3 kaže zbirko nekaterih ikon, ki predstavljajo objekte. Lastnosti objekta lahko enostavno predstavimo, če je vsaka vrednost predstavljena v pravilni grafični obliki. To so lahko na primer debelina črt, velikost znakov, električnih simbolov, razred uporov (v barvah) in drugo.

Ukaze nad objekti lahko prav tako predstavljamo z ikonami. Pri tem razlikujemo različne načrtovalne principe za izvedbo **ukaznih ikon**.

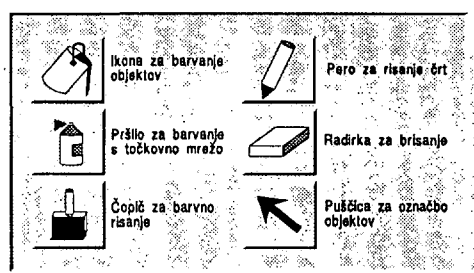


Slika 3: Objektne ikone



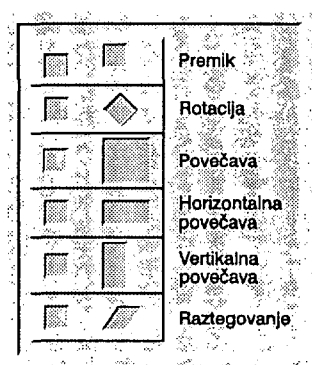
Slika 6: Abstraktna predstavitev ukaznih ikon

Prvi princip je, da z ikono **predstavimo objekt**, ki se pojavi tudi v realnem svetu, na primer, škarje uporabimo za rezanje, vedro barve za polnitev področij, okvirjev, svinčnik za risanje črt in drugo. Slika 4 kaže zbirko takšnih ukaznih ikon. Te ikone se uporabnik težje nauči, ker je potrebna najprej razpoznavna, kaj ikona pomeni in kaj lahko z njo naredi. Zaradi dvokoračnega razumskega procesa je razumevanje in učenje daljše, kot pri enokoračnem, kjer je potrebno samo razpoznavanje pomena ikone.



Slika 4: Ukazne ikone prikazujejo objekte za izvajanje določene naloge

Naslednji princip načrtovanja ukaznih ikon je **prikaz ukazov pred in za učinkom**, kot je prikazano na sliki 5. Ta princip se uspešno uporablja tam, kjer so predstavitve objektov z ikonami enake, kot nastopajo v programu. Če nastopajo različni tipi objektov, ki s predstavitvijo nimajo nobene zveze, potem lahko uporabnik dobi občutek, kot da nad objekti, s katerimi trenutno razpolaga, ne more izvajati operacij, predstavljenih z ikonami. Zato se te vrste ikon pojavljajo le tam, kjer so tipi objektov enaki ali vsaj podobni objektom, prikazanih v ikonah.



Slika 5: Ukazne ikone, ki predstavljajo geometrične pretvorbe

Naslednji princip je **abstraktna predstavitev ukaza**. Tipičen primer je prikazan na sliki 6. Predstavitev je odvisna od različnih kultur in znanja v okolju, kjer se pojavljajo. Znak X lahko, na primer, nekje pomeni tudi brisanje.

Za uporabo ikon lahko rečemo, da je zelo primerna za grafične uporabniške vmesnike, kjer nudijo predstavitev množice objektov na zelo majhnem prostoru. Če so ikone skrbno izdelane, je lahko primerjanje in odčitavanje pomena ikon lažje, kot branje množice ukazov.

## 2.4 Multimedialni dialog

Interaktivne videosisteme, ki so povezani z ostalimi mediji, kot so zvočni vhod/izhod imenujemo **multimedialne dialogne sisteme**. Razvoj teh sistemov imajo nekateri za revolucionarnega<sup>6</sup>. Pri multimedialnih sistemih komunicira uporabnik z računalnikom s pomočjo video slik ali filma, ki mu predstavljajo čim bolj realno sliko delovnega okolja, in s pomočjo zvočnih signalov. Človek lahko, na primer, pri opazovanju procesa na zaslonu, predstavljenega z video sliko, poseže v proces z izgovarjanjem ukazov ali s pomikom kazalca na ustrezno mesto v procesu. To mesto lahko opazuje z dodatno kamero, kar mu poveča preglednost in ustrezno reagiranje. Na koncu lahko celoten uporabnikov postopek še enkrat preverjamo.

S pomočjo digitalnega shranjevanja in obdelave video slike ali zaporedja teh slik in shranjevanja zvočnih signalov se zato realni model (metafora) najbolj približa sliki realnega sveta.

Predpogoj za uporabo takšnih sistemov so zelo zmogljivi računalniki in uporaba optičnih pomnilnikov. Poleg tega so potrebni še zelo hitri postopki za kompresijo in dekompresijo zaporedja video slik v realnem času. V CD-ROM (Compact disk read only memories) pomnilnikih s kapaciteto približno 650 Mbytov se sicer lahko shrani 300.000 strani teksta, vendar samo 30 s dolg film, sestavljen iz video slik. S kompresijo slikovnih podatkov lahko dosežemo s temi CD-ROM-i kapaciteto za približno 72 min filma.

Drugi elementi, ki so potrebni za multimedialne dialoge, so **orodja** za avtorje dialoga, s katerimi načrtujejo in vodijo video in slišne podatke. Poleg tega so potrebni postopki za obdelavo slik, s katerimi lahko iz posnetih video slik izdelamo druge želene poglede na slike. Na MIT (Massachusetts Institut of Technology) razvijajo poleg teh orodij tudi različne primere, kot so interaktivna dokumentacija (razvoj mest), učno okolje (tuji jeziki), komunikacije (raziskovalne kooperacije) in drugo<sup>7</sup>.

## 3. Ostali slogi uporabniških vmesnikov

Slogi, ki smo jih dosedaj opisali, pripadajo slogom, izvedenim za **grafične uporabniške vmesnike**. Ostali slogi sicer niso specifično namenjeni grafičnim uporabniškim vmesnikom, vendar jih kljub temu lahko uporabljamo v grafičnih uporabniških vmesnikih. Ti slogi so:

- **menuji**,
- **ukazna vrstica**,
- **naravni-pogovorni dialog**,
- **dialog s vprašanji in odgovori**.

### 3.1. Menuji

Menuji so najbolj razširjeni slog tako v grafičnih kot v negrafičnih programih. Osnovna prednost menujev je v tem, da uporabnik pri delu z njimi lahko uporablja samo kratkotrajni spomin, ker ima na voljo slike (tekstovni ali ikonski ukazi), ki mu pomagajo pri razpoznavi. To je ravno nasprotno pomnjenju posebnih ukazov, ki si jih mora uporabnik vtisniti v dolgotrajni spomin. Zato so menuji še posebej primerni za začetnike, ki ne potrebujejo velikega naprezanja pri učenju dela s programom. Po drugi strani menuji nudijo tudi zmanjševanje potrebnih ukazov, česar drugi slogi niso zmožni. Z zmanjševanjem potrebnih ukazov mislimo na izbiro samo nekaj možnih ukazov iz celotnega nabora ukazov. Možni ukazi so predstavljeni s črno barvo na svetlem ozadju ali z belo na temnem ozadju, ostali, ki jih ne moremo aktivirati, pa s sivo.

### 3.2. Ukazna vrstica

Uporaba ukazne vrstice je najstarejši način komunikacije z računalnikom. Ta slog lahko uvrstimo tudi med dialoge z iniciativo uporabnika, med katere prištevamo tudi menu. Uporabnik pri uporabi ukazne vrstice najprej vpiše besedo, kodo ali kakšen drug ukaz. Pri tem računalnik takoj izvede akcijo, ki ustreza vnosu. Program vsebuje svoj lastni programski jezik, ki vsebuje veliko množico ukazov in ki jo lahko poljubno razširjamo. Tega jezika se mora uporabnik temeljito naučiti. Vrstica, kamor uporabnik vpišuje ukaze, običajno ne vsebuje nikakršnih dodatnih informacij. V nekaterih primerih pa lahko vsebuje **pozivni stavek** (prompt), ki ga poda uporabniku. Nato uporabnik glede na sporočilo ustrezno odgovori. Ta slog dialoga ima malo različnih oblik. Glavna razlika je v tipu informacije, ki jo uporabnik vpiše in se interpretira v računalniku. Ti tipi so lahko:

- besede (predstavljajo imena funkcij, ki se naj izvedejo),
- črke, okrajšave,
- logični in matematični izrazi,
- akcijske kode (kratke kombinacije različnih ukazov).

Programi z ukazno vrstico ne vsebujejo mnogo zahtev, kaj naj uporabnik naredi. Z enostavnim vtipkanjem ukazov dobi izkušeni uporabnik občutek prostosti in neomejenosti, saj nudijo takšni uporabniški vmesniki visoko stopnjo prilagodljivosti. To pomeni, da lahko ukaze med seboj poljubno križamo in pri tem niso potrebna točno določena zaporedja ukazov.

**Prednosti** te vrste programov so:

- enostavna razširitev ukazov,
- splošna uporabnost vseh funkcij v vsakem trenutku,
- pri rutinskem delu skrajša čas reševanja problemov.

**Slabosti** pa so:

- čas učenja za delo s programom je mnogo daljši, še posebej pri veliki množici ukazov,
- večja možnost napak,
- samo redno, intenzivno delo je učinkovito,
- primeren je samo za izkušene uporabnike, ki znajo hitro tipkati.

### 3.3. Naravni, pogovorni dialog

Naravni, pogovorni dialog je najpomembnejši medij komunikacije med ljudmi. Ta dialog je pogosto prikazan kot alternativa za interaktivne sisteme. Če bi računalnik lahko razumel naše izgovorjene ukaze, bi ga lahko znal uporabljati praktično vsak uporabnik. Najprimernejši bi bil med drugim za slepe ljudi. Vendar je danes izdelava vmesnika v naravnem jeziku še vedno drzno dejanje.

Današnji razpoznavalniki govora z velikim številom besed se morajo posebej prilagajati uporabnikovemu glasu. Če tega posebnega treninga ni, se lahko zelo pogosto pojavijo napake. Največja težava teh razpoznavalnikov je še vedno razpoznavanje hitrega, tekočega govora zaradi nenatančnih premorov med posameznimi besedami. Naravni, pogovorni dialogi so primerni za:

- dialogne sisteme s vprašanji in odgovori,
- poizvedovanje v podatkovnih bazah,
- kombinirani govorni in grafični sistemi.

Zelo primeren je naravni dialog v kombinaciji z direktno manipulacijo. V naravnem dialogu se velikokrat pojavlja dialog, pri katerem pokažemo na neko stvar in povemo, kaj je ali povemo, kaj bi radi z njo. Prav tako lahko z miško izberemo želeno ikono in izgovorimo ukaz, ki manipulira z njo. Z raziskavami so prišli do ugotovitve, da na ta način lahko uporabniki delajo do 60% hitreje kot pri klasični direktni manipulaciji.

### 3.4. Dialog s vprašanji in odgovori

Dialog s vprašanji in odgovori prav tako kot menuji spada med dialoge z iniciativo računalnika in je izmed vseh slogov najbolj preprost. Program postavi vprašanja, na katere uporabnik odgovori z enim od možnih odgovorov. V najpreprostejših oblikah so ti odgovori lahko samo "DA" ali "NE". Dialog je zelo primeren za **binarne izbire**. Primer takšnega procesa najdemo v elektroniki in medicini za reševanje problemov binarnih operacij in postavljanje diagnoz. Na podlagi odgovorov program išče po različnih poteh, dokler ne pride do končnega rezultata ali diagnoze. Prav tako to tehniko srečamo pri vpisovanju formularjev ali različnih podatkov, kot so datumi, letnica rojstva, številka stanovanja in drugo, kjer so prostori za vpis oblikovani tako, da uporabnik točno ve, kaj mora vpisati v posamezen prostor.

Tabela 1 nam kaže primerjavo med najbolj znanimi dialognimi slogi glede na čas učenja, hitrost uporabe, pojavnost napak, razširljivost in ali je potrebno znanje tipkanja.

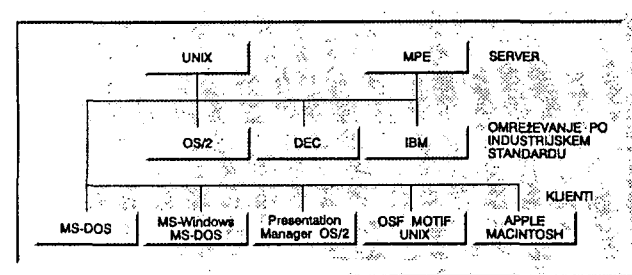
## 4. Zaključek

V zadnjem času se torej na različnih računalniških sistemih vse bolj uveljavljajo **grafični uporabniški vmesniki**. Podobno, kot je potekal razvoj televizije, poteka sedaj tudi razvoj računalniških zaslonov. Na začetku so bili na voljo samo črno-beli računalniški zasloni z zelo majhno resolucijo in brez grafike. V prihodnosti bo razvoj, tako kot pri televiziji, popolnoma izpodrinil črno-bele zaslone in se bodo uveljavili visoko resolucijski zasloni z zmogljivo grafiko. Vsi večji računalniški sistemi, ki so danes uveljavljeni, razvijajo grafične uporabniške vmesnike. DEC uporablja DEC-Windows, IBM-DOS ima Windows, OS/2 uporablja Presentation Manager in UNIX Motif ali Open Look. V prihodnosti se bodo morali vsi ti različni uporabniški vmesniki na nek način med seboj povezati, da se bodo lahko podatki uporabljali tako v enem kot v drugem uporabniškem vmesniku. To velja še posebej, kadar različne sisteme želimo povezati med seboj v mreže. Ker v večini uporabniških vmesnikov manipuliramo z objekti, bo potrebno natančno določiti mrežne objekte in manipulacije nad njimi v mreži. Slika 7 kaže primer povezave različnih uporabniških sistemov v mrežo.

V bližnji prihodnosti bi moral računalnik imeti naslednje lastnosti:

- biti čim bolj "oseben",
- nuditi bo moral uporabo kakršnegakoli programa, ne glede na tip računalniškega sistema,
- uporabljati bo moral mrežo,
- vključiti elektronsko pošto, glasovno pošto in FAX,
- dostop do različnih podatkovnih baz po celem svetu.

Prvi koraki v tej smeri so narejeni s pomočjo grafičnih uporabniških vmesnikov in s poskusi po standardizaciji njihovih slogov. Novejši grafični uporabniški vmesniki vsebujejo objekte, s katerimi enostavno manipuliramo. Prvi večji napredek je tudi uvedba **OLE** (Object Linking and Embedding), ki bo v prihodnosti vse bolj pogost. Tu gre predvsem za to, da lahko podatke, besedila, slike, ki smo jih naredili v poljubnem programu enostavno prenesemo v kakšen drug program. Poleg tega to tudi pomeni, da vse pomembnejše nastavitve, kot so uporaba perifernih enot ali različnih tipov črk, enako uporabljamo ne glede na program. Na ta način se bistveno poenostavi delo z računalnikom in poveča ustvarjalni učinek človeka ob računalniku.



Slika 7: Prikaz mrežne povezave z različnimi računalniki

## LITERATURA

- 1 Fährnich, K.P., "Software-Ergonomie", (State of the Art; 5), München; Wien: Oldenbourg, 1987
- 2 Thimbleby H., "User Interface Design", ACM Press, Addison-Wesley Publishing Company, New York, 1990
- 3 Foley J.D., A. van Dam, S.K. Feiner, J.F. Hughes, "Computer Graphics - Principles and practice", Addison Wesley Publishing Company, New York, 1990
- 4 Shneiderman B., "Direct Manipulation: A Step beyond Programming Languages". IEEE Computer, 1983, 16, str.57-69
- 5 Ziegler J.E., "Direct Manipulation Techniques for the Human- Computer Interface", Eurographics '90 Technical Report Series, Tutorial Note 11.
- 6 Fox,E.A., "The coming revolution of interactive digital video", Communications of the ACM, 32(7),1989,794-801
- 7 Mackay,W.E., Davenport G., "Virtual Video Editing in Interactive Multimedia Applications", Communications of the ACM, 32(7),1989, 802-810.

|                 | WYSI WYG | Direktna manipulacija | Menuji      | Vpis v okvirje | Ukazna vrstica | Naravni dialog | Dialog s vprašanji-odgovori | Multimedialni dialog |
|-----------------|----------|-----------------------|-------------|----------------|----------------|----------------|-----------------------------|----------------------|
| Čas učenja      | kratek   | kratek                | srednji     | kratek         | dolg           | kratek         | kratek                      | kratek               |
| Hitrost uporabe |          | srednja               | srednja     | velika         | velika         | srednja        | majhna                      | majhna               |
| Pojavnost napak | majhna   | majhna                | majhna      | majhna         | velika         | velika         | majhna                      | majhna               |
| Razširljivost   | majhna   | majhna                | majhna      | srednja        | velika         | velika         | velika                      | velika               |
| Znanje tipkanja |          | ni potrebno           | ni potrebno | potrebno       | potrebno       | ni potrebno    | potrebno                    | ni potrebno          |

Tabela 5.1: Primerjava med osmimi različnimi slogi uporabniških vmesnikov