



Analiza domena

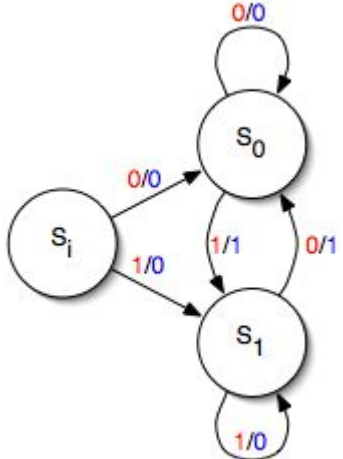
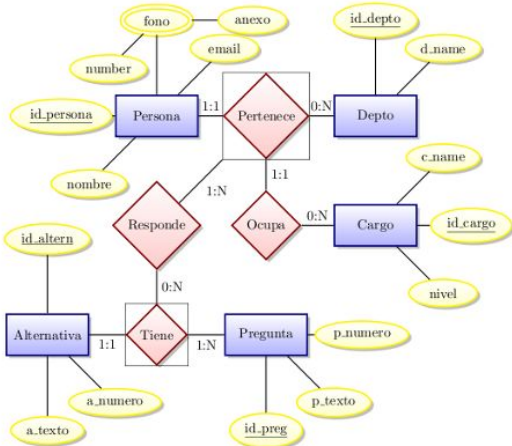
Analiza domena



- Ekspertsko i specijalističko znanje, domen za koji se aplikacija razvija
- Neophodne informacije se mogu dobiti:
 - ispitivanjem eksperata iz datog domena,
 - proučavanjem dokumentacije i literature,
 - **ispitivanjem primarnih korisnika**
- *Zadatak:*

Identifikacija entiteta domena, tj. sve stvari koje su uključene u proces.

- Modelovanje domena



Analiza zadatka

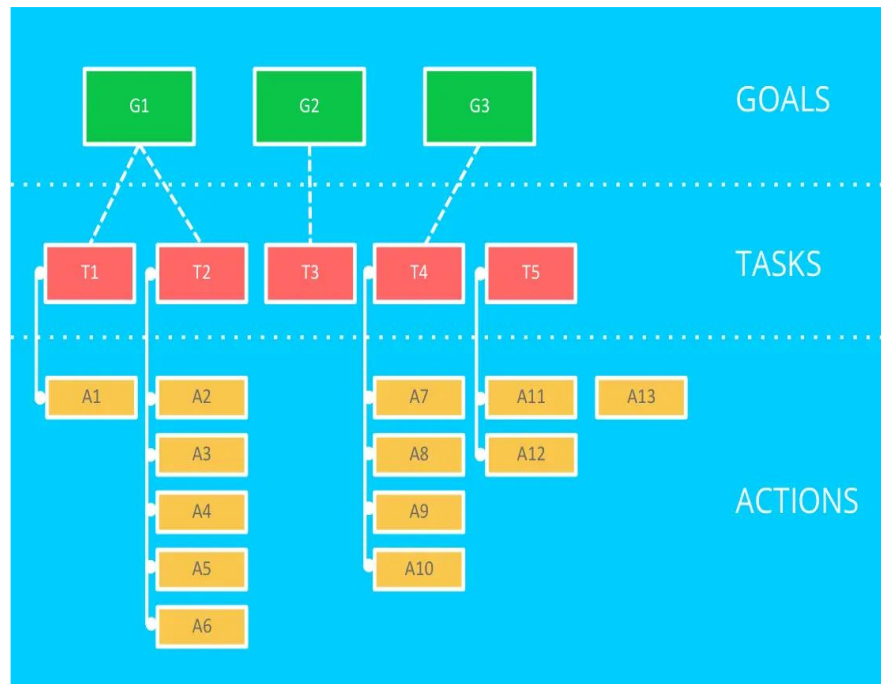


- Kako bi se napravio dobar sistem sa dobrim korisničkim interfejsom, neophodno je odrediti ciljeve koje korisnik namerava da ostvari pomoću tog sistema
- *Zadatak*

Odrediti namenu sistema.

Analiza zadatka

- Posao korisnika možemo opisati kroz:
 - Ciljeve
 - Šta se želi ostvariti?
 - Koji je krajnji rezultat koji korisnik želi da ostvari
 - Zadatke
 - Skup povezanih aktivnosti koje se primenjuju u sekvenci kako bi se ostvario cilj, neki ciljevi se mogu ostvariti na različite načine
 - Akcije
 - Individualna operacija ili korak koji se mora izvesti kako bi se realizovao zadatak



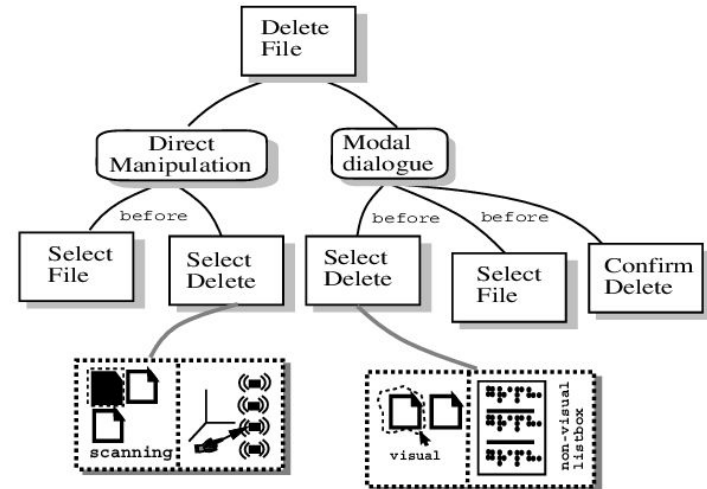
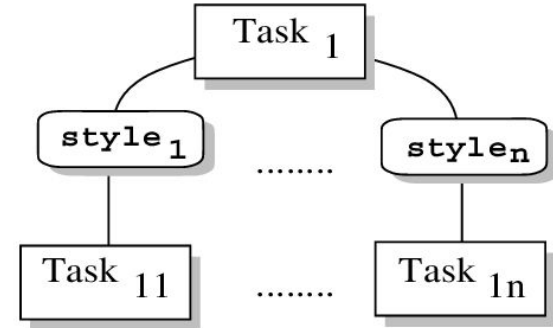
Analiza zadatka



- Zadaci ponekad mogu da se podele na podzadatke
- Da bi se neki ciljevi mogli ostvariti, potrebno je znati i preduslove
 - *Primer: Student da bi prijavio ispit na studentskoj službi prvo mora da se prijavi.*
 - Preduslovi su značajni za dobar dizajn korisničkog interfejsa
 - Omogućuju da se identifikuju situacije koje će dovesti do greške, jer korisnici često ne ispune preduslove
 - Samim tim omogućuju da se izbegnu greške ili da se učine bezazlenim
 - *Primer: Da bi poslali mejl moramo znati adresu primaoca, neki mejl klijenti ne dozvoljavaju klik na dugme Send ako nije uneta adresa primaoca*

Analiza zadatka

- Rezultat: Identifikacija šta treba da se uradi, ali ne i kako
- Hijerarhijska dekompozicija
 - Dobar pristup u analizi zadatka
 - Krene se od osnovnog zadatka sistema, te se on dekomponuje na svoje podzadatke
- Kao i korisnici zadaci imaju svoje karakteristike koje utiču na odluke u izboru i dizajniranju korisničkog interfejsa, kao i na izbor interakcionih uređaja



Analiza zadatka



- Karakteristike taskova
 - Obim u kojem se zadaci razlikuju između jednog i drugog slučaja.
 - Koliko često će zadatak biti obavljen (jednom, jednom dnevno, jednom nedeljno, češće, ...)
 - Znanje i veštine potrebne za obavljanje zadatka.
 - Koliko je rad pogođen promenama okruženja.
 - Da li je vreme kritično za rad.
 - Da li postoje sigurnosne opasnosti.
 - Da li će korisnik raditi sam ili sa drugima.
 - Da li će korisnik obično prelaziti između više zadataka.

Analiza zadatka



Table 4.2 Task Characteristics for Withdrawing Money from an ATM

Does the task vary from one occasion to the next?	No.
How frequently is the task carried out?	May be daily, weekly, or less frequently.
What kinds of skills or knowledge are needed?	Must remember PIN to access machine.
Is the task affected by the environment?	Weather conditions could affect use of machine (e.g., the user may be wearing gloves in winter, it may be raining, bright sunlight may make reading the display difficult).
Is the task time critical?	Users may be in a hurry when using the ATM, since ATMs are often used for their speed and convenience.
Are there any safety or security hazards?	There are no safety hazards in the use of the ATM itself. However, the users' personal safety in relation to onlookers and the safeguarding of their PINs and the cash withdrawn are considerations.
Will the work be done alone or with others?	The work will be done alone.
Will the users normally be switching between several tasks?	Many users will check their balance before withdrawing money. The users will not switch between tasks when withdrawing money, but external factors (like children) may divert their attention.

Analiza zadatka

- Do nekog cilja može da se dođe na više načina.

Table 4.3 Different Task Sequences for Sending a Letter

How Bill Smith sends a letter	How Brenda Jones sends a letter
Write the letter.	Get an envelope.
Get an envelope.	Address the envelope.
Address the envelope.	Write the letter.
Put a stamp on the envelope.	Put the finished letter in the envelope.
Put the finished letter in the envelope.	Put a stamp on the envelope.

Analiza zadatka



- Do podataka o zadacima se dolazi intervjuom i observacijom korisnika.
- Pitanja o zadacima:
 - Gde se izvršava zadatak?
 - Kako je okruženje u kojem se izvršava zadatak?
 - Koliko se često izvodi zadatak?
 - Koji su vremenski okviri za realizaciju zadatka?
 - Koji su resursi na raspolaganju a potrebni da se izvede zadatak?
 - Na koji način se stiče znanje potrebno da se realizuje zadatak?
 - Šta sve može poći po zlu?
 - Ko još učestvuje u zadatku?

Greške u analizi zadatka



- Razmišljanje iz aspekta sistema a ne korisnika
 - “Obavestiti korisnika o sastanku” - zahtev
 - “Obavesti me o sastanku” - zadatak
 - Nekada je samo stvar semantike, ali je često fokus na to šta sistem može, a ne šta korisnik želi
- Suviše rano se fokusira na dizajn interfejsa
 - “Sistemska zvono će obavestiti korisnika ...” - pre detaljno i zavisi od implementacije
- Beleži se šta korisnik zna (konkretno), a ne šta radi (esencijalno)
 - Čuvanje fajla na disk umesto postaram se da je moj rad sačuvan
- Dupliranje loše prakse iz drugog softvera
- Ignorisanje dobrih stvari iz prakse i njihovo neprenošenje u sistem

Kako do bolje analize korisnika/zadatka

- Koncentrisati se na pitanja **zašto** (cilj) i **kako** (podzadaci), a ne samo na šta (jer je korisnicima teže da se fokusiraju samo na šta)
 - Apstrakcija analiza, ali i detaljnija (preciznija)
- Tražiti slabosti i probleme u tome kako korisnici ostvaruju ciljeve i rade svoje zadatke
 - Koji zadaci često rezultuju neuspehom? Zašto?
 - Koji su zadaci nevažni a oduzimaju previše vremena?
 - Koji zadaci nerviraju korisnika? Kako bi korisnici to popravili?



Kako do bolje analize korisnika/zadatka

- Istraživanje konteksta korisnikovog rada
 - Observacija rada korisnika u realnom okruženju
 - Konkretna pitanja
 - Uspostavljanje “*master-apprentice*”
 - Korisnik pokazuje “Kako” i priča o tome
 - Intervjuista gleda, hvata beleške i postavlja pitanja
 - Dovođenje u pitanje pretpostavke i stalno preispitivanje
- Participatory Design (uključivanje svih učesnika u dizajn korisničkog interfejsa)
 - Uključiti reprezentativne korisnike u sam dizajn
 - *Primer: Kako napraviti aplikaciju za knjigovodstvo bez konsultacije sa iskusnim knjigovođom*



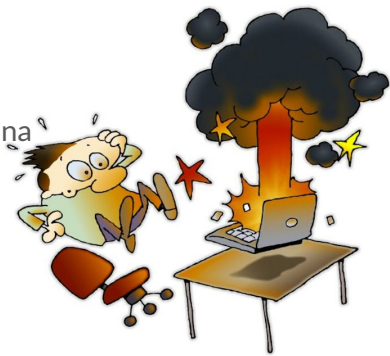
Faktori u formiranju zahteva

- Tokom razvoja računarskih sistema uvek će postojati ograničenja i kompromisi
 - Cena razvoja/budžet/rokovi
 - Dostupna tehnologija i povezanost sa hardverom i drugim softverom
 - Stavovi/motivi individualnih zainteresovanih strana
 - Kontradiktorni zahtevi
 - Organizaciona politika



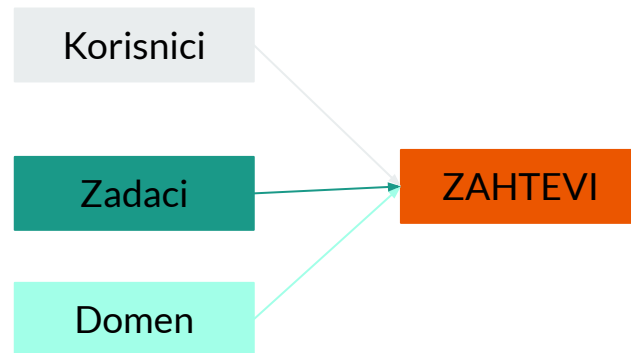
Problemi tokom formiranja zahteva

- Velika lista ograničenja i problema, na neke se ne može uticati
 - Nedovoljan broj korisnika/zainteresovanih strana, nekompletna lista zahteva
 - Loša evidencija zahteva i promena funkcija usled otkrivanja novih zahteva, može dovesti do ispuštanja nekog zahteva ili pogrešne interpretacije
 - Loša koordinacija faze prikupljanja zahteva
 - Loša komunikacija između korisnika na različitim nivoima odlučivanja, loša komunikacija sa programerskim timom
 - Određivanje domena nije uopšte lako i trivijalno, greške mogu realizovati pogrešnim zahtevima
 - Ljudi koji razumeju problem možda nisu dostupni ili zainteresovani
 - Organizacioni i politički faktori
 - Zainteresovani učesnici ne znaju šta žele od sistema
 - Okruženje se stalno menja kao i ljudi na pozicijama, te zahtevi mogu zastareti usled tih promena
 - Prenaglano zaključivanje (opis idealnog studenta, a ne kakav je zaista)



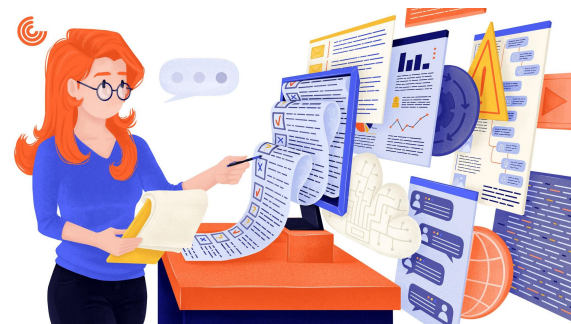
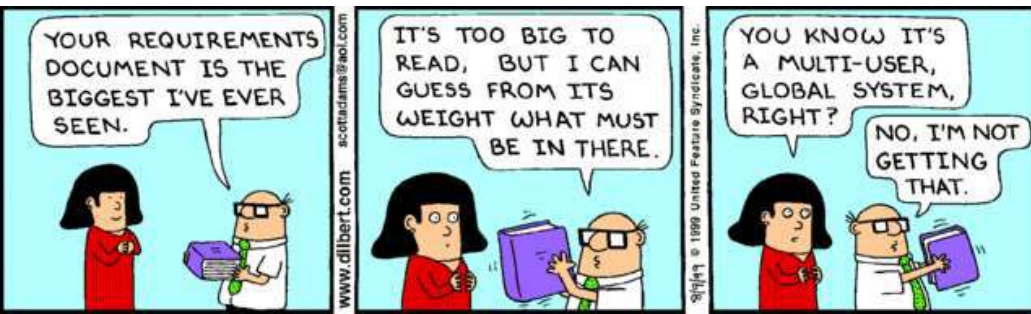
Formiranje zahteva

- Zahtevi vezani za interfejs su deo zahteva čitavog sistema
- Trebaju da obezbede opis onog šta sistem treba da uradi, ne i kako
- Analizom skupljenih podataka iznose se zaključci koji se beleže u dokument (specifikaciju) zahteva
- Dokument zahteva:
 - a. arhitektonski zahtevi (potrebe arhitekture sistema)
 - b. funkcionalni zahtevi (funkcionalnost koja se mora izvršiti)
 - c. nefunkcionalni zahtevi (karakteristike sistema kao zahtevi kvaliteta)
 - d. zahtevi ograničenja (margina rada sistema)



Formiranje zahteva

- Ne postoji standardni način da se napiše dokument zahteva
 - Zahtevi koji se odnose na karakteristike korisnika
 - Zahtevi koji se odnose na zadatke i njihove karakteristike
 - Usability zahtevi
 - Zahtevi koji se odnose na domen i radno okruženje
 - Limitirajući faktori, ograničenja i preduslovi
 - ...
- Piše se prirodnim jezikom koji može svako razumeti, jer će ga čitati ljudi različitog profila
 - Govorni jezik može predstavljati i problem jer se može interpretirati na više načina



Formiranje zahteva - potrebno znanje



- Na kvalitet, sveopštost i tačnost dokumenta zahteva, pored informacija o korisniku, zadacima, domenu, potrebno je i znanje iz samog dizajna korisničkog interfejsa
 - Znanje iz teorije, kognitivne psihologije i delom iz iskustva koje dolazi proučavanjem dobrog korisničkog interfejsa
- Upotreba specifičnog i specijalizovanog znanja, formulisanog kroz:
 - smernice (abstraktna uputstva visokog nivoa opštosti koje zahtevaju interpretaciju kako bi se mogle primeniti) i
 - pravila (detaljna, specijalizovanja uputstva niskog nivoa opštosti koja se mogu pratiti uz minimalno interpretacije)
- Treba uvek primenjivati obazrivo i u skladu sa kontekstom - uzimati u obzir korisnike, njihove zadatke i domen

Četiri psihološka principa



1. Korisnici vide ono što očekuju da vide.

Ako su dugme OK i Cancel na jednom dijalogu na jednoj poziciji, a na sledećem su na drugoj poziciji, korisnik na drugom dijalogu može videti da je OK na mestu sa prvog dijaloga.

2. Korisnici se fokusiraju samo na jednu stvar.

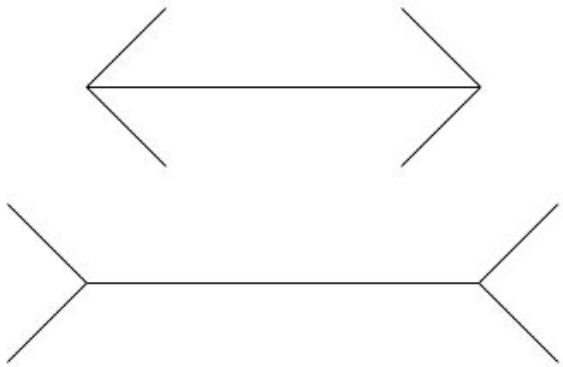
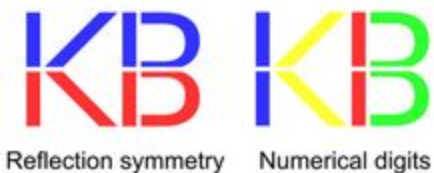
U bučnom okruženju (velika gužva) korisniku može da odluta pažnja, korisnički interfejs treba da ga podseća na to šta treba da uradi sledeće

3. Lakše je opaziti strukturirani razmeštaj kontrola (layout).

Teško je uočiti gde treba kliknuti, ako korisnički interfejs ne daje nagoveštaje gde se može kliknuti (linijama, bojom, senkom).

4. Lakše je prepoznati nešto nego se podsećati.

Daleko je lakše prepoznati nešto sa menija ili dugmeta, nego se prisećati komande.



THE CAT

student home

Welcome to the **Learner's Guide** to the **OU**

[Feedback](#)

The Learner's Guide will help you make informed decisions about your study.
 Welsh: Am fanylion y gwasanaethau sydd ar gael drwy gyfrwng y Gymraeg, [cliciwch yma](#).

Course Choice How to choose a course or qualification.

Services for disabled students Advice for disabled students and others who have additional needs.

Career Planning Helpful information on career development and planning.

Learning Skills A range of resources to help improve your study skills.

For OU students only: Learning with the OU

These sites have more resources to help you get the most out of your studies, including study skills activities, help with completing assignments, advice on personal and career development.

[TOP](#)

[Site feedback \[Cymraeg\]](#) | [Site accessibility](#)
[Disclaimer](#) | [Privacy policy](#)
 Copyright © 2004 The Open University

Destination	Flight	Carrier	Depart	Arrive	Rates	
					Business	Standard
Aberdeen	4171	BA	0845	0945	£155	£102
Dublin	664	FR	1035	1135	£149	£100
Toulouse	8064	AF	1110	1410	£307	£182
Frankfurt	4618	LH	1115	1355	£222	£152
Amsterdam	2045	UK	1130	1335	£222	£152
Copenhagen	8363	BA	1145	1445	£315	£187
Paris-CDG	1803	BA	1150	1400	£248	£165
Exeter	446	JY	1205	1305	£155	£102
Glasgow	1903	BA	1210	1310	£155	£102
Munich	4526	LH	1225	1525	£301	£179
Geneva	8413	BA	1235	1420	£222	£152
Aberdeen	4172	BA	1245	1345	£155	£102

Dest: Copenhagen (BA8363) Dep: 1145; Arr: 1445
(B/S: £315/187)

Dest: Paris-CDG (BA1803) Dep: 1150; Arr: 1400
(B/S: £248/165)

Dest: Exeter (JY446) Dep: 1205; Arr: 1305
(B/S: £155/102)

Dest: Glasgow (BA1903) Dep: 1210; Arr: 1310
(B/S: £155/102)

Dest: Munich (LH4526) Dep: 1225; Arr: 1525
(B/S: £301/179)

Dest: Geneva (BA8413) Dep: 1235; Arr: 1420
(B/S: £222/152)

Dest: Aberdeen (BA4172) Dep: 1245; Arr: 1345
(B/S: £155/102)

Formiranje zahteva: Principi iz iskustva

- Vidljivost
 - Da li su kontrole vidljive, jasne?
- Očiglednost, dostupnost
 - Da li je očigledno kako se kontrole koriste?
- Obaveštenje
 - Da li je jasno iskazana reakcija sistema na korisničku akciju?

Visibility, Affordance, and Feedback

The controls on newer VCRs are generally more visible than on older models. There are good mappings between the controls and their effects — that is, the functions provided by the VCR interface map onto or are related to the user's goals and needs. Controls often have just one function; for example, there may be several push buttons that do only one thing, like an on/off button that simply turns the VCR on or off. There is affordance, as the buttons make it obvious how they are operated (i.e., they afford pressing). There is feedback on the display, and the system is generally understandable. In general, the relationships between the user's goals, the required actions, and the results are sensible, meaningful, and not arbitrary.



Figure 5.7 Door handles. The one on the left affords pulling. The one on the right affords levering. What about the one in the middle?