PROJEKTIRANJE INFORMACIJSKIH SUSTAVA
2. KOLOKVIJ

1. Koji je prvi korak u fazi dizajna? Koje su tri mogućnosti u prvoj fazi dizajna te koje se prednosti, a koji nedostatci svake od tri mogućnosti?

Prvi korak u fazi dizajna je odabir strategije izrade sustava:

a) Razvoj vlastite (custom) aplikacije (iz temelja) in-house:

Prednosti: omogućava fleksibilnost i kreativnost, uklopiv je sa postojećim tehnologijama i standardima, razvija tehničke i funkcionalne vještine tima

Nedostaci: zahtjeva duže vrijeme izrade i veći trud, problemi sa velikom dokumentacijom i prevelikim zahtjevima klijenta, nedostatak poznavanja novih tehnologija, veći troškovi, veliki rizik

b) Kupovina gotove (packaged) aplikacije i prilagodba klijentovim zahtjevima:

Prednosti: dostupan za mnoge uobičajene poslovne potrebe (knjigovodstvo, skladištenje), gotove aplikacije su dobro testirane, pouzdane, ušteda vremena i novca

Nedostaci: Rijetko u potpunosti odgovara potrebama klijenta

c) Razvoj aplikacije izvan kompanije (outsourcing)

Mogući rizici: gubitak povjerljivih informacija, gubitak kontrole nad daljnjim razvojem, gubitak mogućnosti učenja

2. Što se uzima u obzir prilikom odabira strategije dizajna?

– Poslovne potrebe
– "In-house" iskustvo
– Znanje o projektima
– Upravljanje projektima
– Vremenski okvir

3. Što je alternativna matrica pri odabiru strategije dizajna? Kako se matrica razvija (na temelju kojih informacija)?

– Kombinacija nekoliko analiza provedivosti u jednoj matrici
– Uključiti tehničku, financijsku i organizacijsku izvedivost
– Dodijeliti značaj koji će indicirati relativnu važnost kriterija
– Dodijeliti proračune koji će indicirati odnos alternative i kriterija

4. Što je arhitektura softverskog sustava (definicija)?

Arhitektura softverskog sustava je struktura ili strukture sustava, koje sadrže softverske elemente, vanjska vidljiva svojstva tih elemenata i njihove međusobne relacije.

5. Što su vanjska vidljiva svojstva elemenata u arhitekturi softverskog sustava?

Vanjska vidljiva svojstva elemenata podrazumijevaju servise koje elementi nude drugim elementima (element dohvaća podatke iz baze, element izračunava srednju vrijednost), performanse elemenata, dijeljenje resursa sa drugim elementima i sl.

6. Što su ADL (architecture description languages) jezici i čemu služe? Ako znate neke nabrojite ih.

ADL jezici pružaju formalni način za predstavljanje arhitekture.

– Mogu podržavati automatsko generiranje koda.
– Omogućavaju analizu arhitekture kroz analizu performansi, uporabljivosti (usablitiy) sa aspekta korisnika, sigurnosti i sl.
– Namijenjeni su interpretaciji i od čovjeka i od računala.
– Nažalost ne postoji suglasnost što ADL jezici trebaju predstaviti u sustavu, posebno što se tiče ponašanja.
– ADL jezici: ACME (CMU/USC), Rapide (Stanford), Wright (CMU), Unicon (CMU),…

7. Napišite sve što znate o IEEE 1471 standardu (Recommended Practice for Architecture Description of Software-Intensive Systems).

Standard za predstavljanje arhitekture softverskog sustava. Opis arhitekture sustava se koristi za:

– Predstavljanje sustava i njegovu procjenu
– Komunikaciju među zainteresiranima za sustav
– Procjenu i usporedbu arhitektura
– Planiranje i upravljanje razvojem sustava
– Predstavljanje trajnih karakteristika i principa sustava za olakšavanje mogućih promjena sustava (za web aplikaciju ne možemo unijeti element koji se odnosi na desktop aplikaciju)
– Verifikaciju implementacije sustava sa postavljenom
Arhitektura se prema IEEE 1471 organizira kroz različite prikaze, analogne nacrtima u građevinarstvu (vodovodne instalacije, statika zgrade, …). Definirani prikazi su:
– Functional/logic view
– Code/module view
– Development/structural view
– Concurrency/process/thread view
– Physical/deployment view
– User action/feedback view
– Data view

8. Navedite arhitekturalne predloške koje smo naveli na predavanjima.

– Klijent-server arhitektura (2-slojna, više-slojna, …)
– Troslojna arhitektura (prezentacijska logika, poslovna logika, podatkovna logika)
– Objektno-orijentirana arhitektura
– Servisno orijentirana arhitektura
– Front-end/back-end arhitektura (Joomla)
– Monolitna aplikacija…..

9. Kada govorimo o klijent-server arhitekturi softverskog sustava naveli smo tri tipa servera. Koji su to tipovi i koja svojstva imaju?

– "Mainframe"
  – Velika i izrazito snažna računala
  – Visoka cijena
  – IBM zSeries
– "Minicomputer"
  – Velika računala
  – Cijena reda veličine stotine tisuća dolara
  – IBM, HP, Sun
  – RISC arhitektura
– "Microcomputer" (osobna računala)
  – Mala desktop računala
  – Cijena reda veličine desetke tisuća dolara

10. Koje su prednosti, a koji nedostatci arhitekture zasnovane na serveru (Server-Based Architecture)?

– Jednostavnost implementacije
– Ne zahtjeva komplicirano upravljanje
  – S većim brojem korisnika i usluga, server postaje opterećen
  – Skupa nadogradnja centralnog servera

11. Koje su prednosti, a koji nedostatci arhitekture zasnovane na klijentu (Client-Based Architecture)?

– Rasterećenje servera
  – S većim porastom mrežnih aplikacija mrežna čvorišta postaju opterećena
  – Radi slabog kapaciteta obrade i klijenti postaju opterećeni

12. Koje su prednosti, a koji nedostatci arhitekture sa ravnopravnim učešćem?

– Skalabilnost (mogućnost nadogradnje)

– Kompatibilnost sa raznim proizvodima i proizvođačima kroz middleware
– Lakše formiranje web-based sustava
– Otpornost na kvarove
  – Kompleksnost
  – Novi programski jezici i tehnike
  – Kompleksnije osvježavanje

13. Na temelju čega se radi izbor arhitekture softverskog sustava?

– Proširiti ne-funkcionalne zahtjeve
– Na osnovu detaljnih ne-funkcionalnih zahtjeva odabrati prikladnu arhitekturu

14. Što obuhvaća specifikacija softvera i hardvera?

– Određuje softverske potrebe
  – OS, posebne softverske potrebe
  – Obuka, garancija, održavanje, kupnja licenci
– Određuje hardverske potrebe
  – Server(i), klijenti, periferni uređaji, backup uređaji, uređaji za pohranu
  – Minimalna konfiguracija

15. Što je korisničko sučelje, a što sistemsko sučelje softverskog sustava?

– Korisničko sučelje je dio sustava preko kojeg korisnik komunicira sa sustavom.
– Sistemsko sučelje je dio sustava preko kojeg sustav komunicira sa drugim sustavima (npr. sustav treba eksportirati podatke za neki drugi sustav).

16. Koja su dva osnovna tipa korisničkog sučelja?

– Grafičko korisničko sučelje (Graphical user interface (GUI)) koristi prozore, ulazni uređaj miša, izbornike itd.
– Tekstualno sučelje (Text-based interface) se također koristi npr. na mainframe računalima ili Linux OS-ovima.

17. Koja tri osnovna dijela uključuje korisničko sučelje? Ukratko opišite čemu služe ta tri dijela korisničkog sučelja.

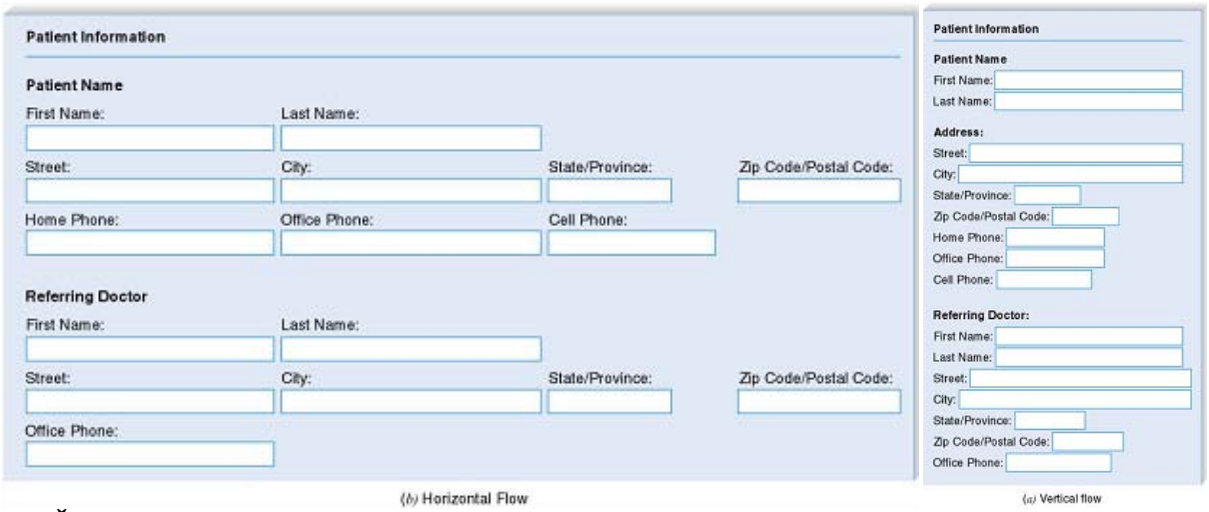
– Mehanizam navigacije (navigation mechanism) određuje na koji način korisnik postavlja zahtjeve prema sustavu tj. kako korisnik kaže sustavu što da radi (npr. korisnik klikne na izbornik da dobije prozor za unos informacija o kupcu i onda klikne na botun da bi spremio te informacije u bazu)
– Mehanizam ulaza (input mechanism) definira način na koji će sustav dobiti informaciju (npr. postoji forma sa 5 polja preko koje korisnik unosi podatke o kupcu)
– Mehanizam izlaza (output mechanism) definira način na koji sustav isporučuje informaciju korisnicima (printani izvještaj, graf,…)

18. Kojih 6 osnovnih principa se koristi kod dizajna korisničkog sučelja?

– Razmještaj (layout)
– Svijest o sadržaju (content awareness)
– Estetika
– Iskustvo korisnika (user experience)
– Konzistentnost
– Minimiziranje korisnikovog truda (minimize user effort)

19. Što je princip razmještaja kod dizajna korisničkog sučelja? Koji se standardni razmještaj danas najčešće koristi? Nacrtajte primjer horizontalnog i vertikalnog razmještaja.

– Razmještaj određuje organiziranje određenih područja određenih funkcionalnosti na ekranu, formi, izvještaju. Informacija korisniku može biti prezentirana u različitim područjima. Ta područja se trebaju koristiti konzistentno kroz aplikaciju.
– Većina softvera danas koristi tzv. Windows i Macintosh



20. Što je princip svijesti o sadržaju kod dizajna korisničkog sučelja i na koji način se podiže svijest o sadržaju?

– Svijest o sadržaju je sposobnost sučelja da korisnik percipira informacije koje sučelje sadrži s minimumom truda.

– Svijest o sadržaju se podiže na način da:

1. Svaki ekran, forma, izvještaj trebali bi imati naslove.
2. Meniji bi trebali pokazivati
   – Gdje ste
   – Odakle ste došli da bi stigli tu
3. Trebalo bi biti jasno koja je informacija unutar kojeg područja (npr. oznaka područja).
4. Oznake polja i oznake područja trebalo bi odabrati pažljivo. Oznake trebaju biti što kraće, a da korisnik preko njih ipak ima svijest o sadržaju područja ili polja. Poljem nazivamo pojedinačni element podataka koji se unose ili prikazuju.
5. Format podataka u polju bi trebao biti točno određen. (Npr. datum formata 10/5/2009 – u nekim zemlja je to 5.10.2009., a u nekima 10.5.2009.)
6. Upotreba datuma i verzije da bi se pomoglo korisnicima (osobito na izvještajima).

21. Koja se estetska pravila koriste za slova kod dizajna korisničkog sučelja?

– Općenito pravilo za dizajn teksta je da sav tekst treba biti u istom tipu slova i slične veličine.

– Slova trebaju biti velika barem 8 piksela, ali preporučena minimalna veličina je 10 piksela.

– Promjena veličine ili tipa slova bi trebala ukazivati na neku promjenu u informaciji (npr. naslovi područja mogu biti veći).

– Tipovi slova poput Times New Roman (tzv. serif fonts) najvidljiviji su u tiskanim izvještajima, osobito ako su slova mala, dok su Arial, Helvetica (tzv. sans serif fonts) vidljivija na ekranu.

– Izbjegavati korištenje velikih slova (PREZIME –> Prezime).

22. Što je princip iskustva korisnika kod dizajna korisničkog sučelja i koja je razlika između početnika i iskusnih korisnika aplikacije?

– Iskustvo korisnika se odnosi na dizajniranje sučelja u skladu sa korisnikovim iskustvom.

– Korisnicima početnicima je bitno koliko je vremena potrebno da se nauče služiti programom (ease of learning).

– Iskusni korisnici su zainteresiraniji za jednostavnost uporabe programa

– Početnici rijetko koriste statusne informacije za razliku od iskusnih korisnika.

– Početnici očekuju da su sve funkcionalnosti predstavljene u izborniku jer to olakšava učenje korištenja aplikacije, dok iskusni korisnici očekuju da su samo često korištene funkcionalnosti predstavljene u izborniku jer to olakšava korištenje aplikacije.

23. Što je princip konzistentnosti kod dizajna korisničkog sučelja?

– Omogućuje korisnicima da predvide što će se dogoditi.

– Kada je sučelje konzistentno korisnik u interakciji sa jednim dijelom sučelja (npr. jednom formom) nauči kako koristiti i druge dijelove sučelja jer su konzistentni. Npr. korištenje iste ikone ili iste naredbe sa određenom funkcijom kroz cijeli sustav (npr. Undo)

24. Što je princip minimiziranja korisnikova truda kod dizajna korisničkog sučelja?

– Korisničko sučelje treba dizajnirati na način da se minimizira trud koji korisnik treba uložiti za obavljanje neke akcije.

– Trud u obavljanju akcije preko korisničkog sučelja se sastoji u korištenju ulaznih uređaja preko kojih korisnik komunicira sa aplikacijom, tj. smanjenje klika mišem i pritisaka tipki tipkovnice. (Većina dizajnera primjenjuje pravilo tri klika)

25. Dizajn korisničkog sučelja radi se u pet koraka. Koji su to koraci? Opišite ih ukratko.

1. razvoj scenarija korištenja – na osnovu DFD-ova i slučajeva korištenja iz faze analize i to je opći pregled koraka da bi izvršio određeni zadatak
2. dizajn strukture sučelja – definira osnovne komponente ili elemente sučelja (ekrane, forme i sl.) i način na koji su komponente sučelja međusobno povezane.
3. dizajn standarda sučelja – temeljni dizajnerski elementi zajednički pojedinim ekranima, formama, i izvještajima unutar aplikacije.
4. prototipiranje dizajna sučelja – to je nacrt izgleda ili simulacije ekrana, formi ili izvještaja.
5. evaluacija sučelja – identificira moguća poboljšanja predloženog dizajna sučelja.

26. Opišite dijagram strukture sučelja (interface structure diagram) koji se koristi za dizajn strukture sučelja. Kako se dobiva i što prikazuje?

– Prikazuje sve ekrane, forme i izvještaje, njihovu povezanost, te način na koji se korisnik kreće između pojedinih ekrana, formi i izvještaja.

– Nalikuju DFD (kvadrati i linije), ali bez standardnih pravila.

– Svaka komponenta sučelja (forma, ekran, izvještaj) se predstavlja kvadratom sa jedinstvenim identifikatorom (brojem) i jedinstvenim imenom (U ISD kvadratima se obično navodi i oznaka procesa koji je vezan uz određeni element sučelja. Određeni procesi mogu biti vezani uz nekoliko elemenata sučelja.)

27. Koje se metode koriste za prototipiranja sučelja? Opišite ih.

1. Storyboard – najčešće se koristi papir za prikaz izgleda sučelja, ali postoji i elektronički storyboard. Elementi sučelja se nacrtaju i onda se kao kod crtanih filmova slažu slike sučelja da bi se pokazalo što se dešava u navigacija ili drugim aktivnostima korisnika.
2. HTML prototip – najčešće se koristi. Sučelje se prikazuje kao HTML stranice. Ovakav prototip omogućava korisniku "stvarno" testiranje sučelja jer prototipovi sadrže i navigaciju i polja za unos i botune. Jedino što nije implementirana funkcionalnost iza sučelja.
3. Jezični prototip (language prototype) – prototip sučelja se razvija slično kao i HTML prototip, ali u onom programskom jeziku u kojem će se razviti i "pravo" sučelje. Razvoj traje duže nego kod HTML prototipova, ali korisnik dobiva točan izgled budućeg sučelja.

28. Koje se metode koriste za evaluaciju korisničkog sučelja? Opišite ih.

1. Heuristička evaluacija – članovi tima uspoređuju razvijeni dizajn sa principima dizajna.
2. Walk-through evaluacija – tim simulira kretanje kroz sučelje korisniku (korištenjem storyboarda ili HTML ili jezičnih prototipova) koji identificira moguća poboljšanja sučelja.
3. Interaktivna evaluacija – korisnici sami iskušavaju sustav (za to su potrebni ili HTML ili jezični prototipovi ili gotovi sustav).
4. Formalno testiranje uporabljivosti – je vrlo skupo jer se provodi u uvjetima specijalnog laboratorijskog testiranja.

29. Navedite i opišite tipove navigacijskih kontrola.

1. Jezik – Korištenjem naredbenog jezika korisnik naredbu unosi posebnim jezikom (Linux shell, SQL,…). Ovaj se pristup danas rijetko koristi osim kada veliki broj mogućih naredbi i njihovih kombinacija čini praktički nemogućim stvaranje izbornika (SQL).

2. Izbornici su najčešći tip navigacijske kontrole. Izbornik korisniku pruža listu mogućih komandi od kojih korisnik izabire. Lakši su za korištenje nego komandni jezici. Podizbornici su zatvoreni dok ih korisnik ne odabere tako da ne pružaju uvid u funkcije koje nude, pa je bolje koristiti prostrani plitki izbornik. Bilo bi dobro da izbornik/podizbornik ima do 8 elemenata. Međutim to nije uvijek moguće pa se to kompenzira grupiranjem elemenata i odvajanjem npr. horizontalnom linijom kako bi se korisnik što lakše snašao. Korištenje shortcut ili hot key tipki je također poželjno. Grupiranje sličnih elemenata izbornika može ići ili po objektu sučelja (npr. klijenti, transakcije,…) ili akciji (novo, izmjena,…).
3. Direktna manipulacija omogućava korisniku direktno pozivanje naredbe na objektu sučelja. Npr. u Windows Exploreru kada povlačimo datoteku sa jednog mjesta na drugo direktno na datoteci izvršavamo naredbu copy, move, delete. Nedostatak direktne manipulacije je što nije uvijek intuitivna tj. korisnik ne može intuitivno zaključiti kako izvršiti neku naredbu.

30. Što je svrha dizajna ulaza korisničkog sučelja i koje dvije smjernice treba poštivati pri dizajnu ulaza?

– Cilj dizajna ulaza je jednostavno i lako uhvatiti točnu informaciju za sustav.
Smjernice:
- – Online procesiranje,
- – Pozadinsko (batch) procesiranje

31. Što je svrha dizajna izlaza korisničkog sučelja i koji su osnovni principi dizajna izlaza?

– Cilj mehanizma izlaza je prezentirati informacije korisniku na način da ih korisnik razumije uz što manje truda.
– Osnovni principi dizajna izlaza su:
- – Razumijevanje upotrebe izvještaja
  - – Referencirajući ili potpuni (cover– to– cover)
  - – U realnom vremenu ili batch izvještaji
- – Racionalno korištenje informacija
  - – Samo potrebne informacije, ništa više
- – Minimiziranje pristranosti (npr. izvještaj sortiran po abecedi stavlja naglasak na podatke s obzirom na abecedu)

32. Što određuje dizajn programa?

Određuje koji se programi trebaju napisati, na koji način se pojedini napisani dijelovi kôda međusobno povezuju kako bismo dobili integralno funkcionalnu aplikaciju, te na koji način programeri trebaju napisati te programe.

33. Koji su koraci prilikom dizajna programa? Opišite ih ukratko.

Koraci prilikom dizajna programa su:
1. Definiranje fizičkog dijagrama toka podataka (physical data flow diagram) koji se izvodi iz dijagram toka podataka dobivenih u fazi analize programa.
2. Definiranje dijagrama strukture (structure chart) povezivanjem svih procesa koji se izvode u sustavu.
3. Izrada detaljnih instrukcija za pisanje programa, programske specifikacije (program specifications).

34. Koja je razlika između logičkog i fizičkog DFD-a? Koja četiri koraka se provode prilikom prebacivanja logičkog DFD-a u fizički DFD?

Razlika između fizičkog i logičkog DFD je u tome što fizički DFD sadrži detalje o izradi sustava (npr. spremište podataka je UTF8 tekstualna datoteka na disku).
Četiri koraka:
1. dodavanje implementacijskih referenci
2. postavljanje granice između čovjeka i uređaja
3. dodavanje sistemskih procesa, podataka i tokova
4. ažuriranje podataka u tokovima

35. Što prikazuje dijagram strukture programa (structure chart)? Opišite dijelove dijagrama strukture programa.

– Dijagram strukture (structure chart) programa prikazuje sve module koji trebaju biti uključeni u program u hijerarhijskom formatu toka izvršavanja programa (gornji modul poziva module na nižoj razini).

– Modul se označava pravokutnikom.

– Linije koje povezuju module označavaju prenošenje kontrole izvršavanja aplikacije od jednog modula do drugog.

– Kontrolni modul (control module) je modul više razine koji sadrži logiku izvršavanja modula niže razine koji se nazivaju podređeni (subordinate) u odnosu na promatrani modul.

– Iteracija se označava zakrivljenom strelicom dok se selekcija označava sa rombom.

36. U koja četiri koraka se odvija izgradnja strukturnog dijagrama?

– Identifikacija modula i razina

– Identifikacija selekcija i iteracija

– Dodavanje sprege

– Provjera strukturnog dijagrama

37. Opišite transakcijsku strukturu strukturnog dijagrama.

Transakcijska struktura se koristi kada svaki modul izvodi zasebnu transakciju. Obično ima malo ulaznih, a dosta izlaznih modula. Obično se nalazi na višim razinama dijagrama strukture. Npr. imamo modul koji sadrži izbornik i ovisno o korisnikovoj akcija preko izbornika se poziva modul koji ima specifičnu funkcionalnost.

38. Opišite transformacijsku strukturu strukturnog dijagrama.

Transformacijska struktura se koristi kada su moduli povezani i zajedno čine neki proces koji transformira ulaz u izlaz. Obično ima dosta ulaznih procesa, a malo izlaznih. Obično se nalazi na nižim razinama dijagrama strukture. Npr. imamo slijed web formi koje moramo popuniti jednu za drugom, svaka web forma je razvijena kao jedan ASP.NET ili PHP modul.

39. Što uključuju mjere dobrog dizajna modula strukturnog dijagrama? Opišite ukratko.

– Kohezija – važno je da pojedinačni modul ne uključuje različitu funkcionalnost jer to otežava izmjene tog modula. Npr. da modul za dohvaćanje postavki korisnika ne uključuje obradu narudžbi.

– Sprega modula – cilj dizajna je postići što veću neovisnost modula. Naravno, moduli trebaju iskazivati određenu međusobnu ovisnost kako bi tvorili cjelinu. Tipičan primjer implicitne sprege su globalni podaci.

– Prikladne razine fan-in i fan-out – fan-in opisuje broj kontrolnih modula koji komuniciraju sa podređenim modulom, fan-out opisuje broj podređenih modula s kojima komunicira kontrolni modul.

40. Što je kohezija modula i koja dva tipa kohezije smo definirali? Pitanje 39 +

Kohezija – važno je da pojedinačni modul ne uključuje različitu funkcionalnost jer to otežava izmjene tog modula. Npr. da modul za dohvaćanje postavki korisnika ne uključuje obradu narudžbi.

Tipovi kohezije:

– Slabiji oblik kohezije naziva se logička kohezija

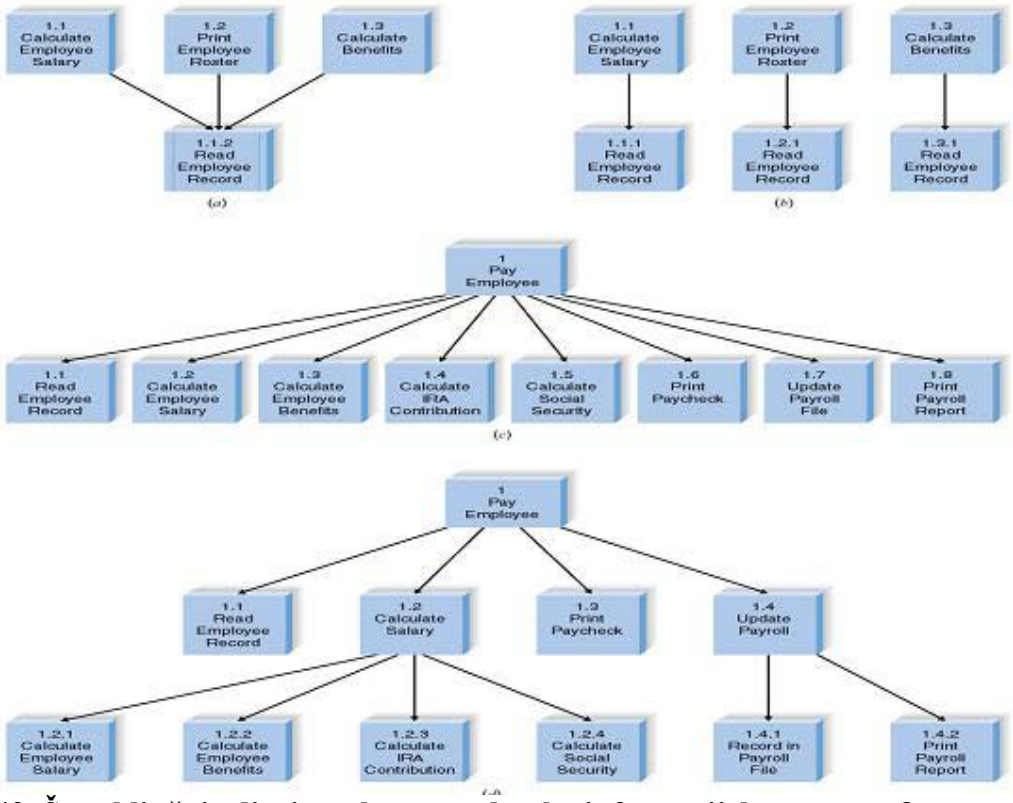
– Jači oblik kohezije naziva se funkcionalna kohezija

41. Što je implicitna sprega modula?

Implicitna sprega (implicit coupling) modula nije očigledna, a često uzrokuje greške u softveru. Tipičan primjer implicitne sprege su globalni podaci, većina programskih jezika više razine podržava primjenu globalnih podataka. Često se promjene na globalnim podacima dešavaju kao popratna pojava (side effect) izvođenja modula.

42. Što je fan-in i fan-out modula strukturnog dijagrama i koje su prikladne razine fan-in i fan-out modula?

Fan-in opisuje broj kontrolnih modula koji komuniciraju sa podređenim modulom, fan-out opisuje broj podređenih modula s kojima komunicira kontrolni modul.



43. Što uključuje dizajn pohrane podataka informacijskog sustava?

Dizajn pohrane podataka obuhvaća definiranje:

– formata pohrane podataka,

– formata podataka (slogova, redaka) u pohrani,

– optimizaciju pohrane podataka (brzine i veličine).

44. Koja su dva osnovna tipa formata pohrane podataka?

– datoteke,

– baze podataka.

45. Opišite ukratko četiri različita tipa baza podataka koje smo definirali na predavanjima.

1. Naslijeđene (legacy) aplikacije – obično koriste baze podataka koje se temelje na starijim tehnologijama i danas se više ne koriste u razvoju aplikacija, ali postoji veliki broj sustava koji se temelje na njima. Primjeri su hijerarhijske baze i mrežne baze

2. Relacijske baze podataka – podaci su u relacijskoj bazi grupirani po tablicama. Relacijska baza je kolekcija tablica koje sadrže slogove. Tablice sadrže primarni ključ (primary key) kao polja u slogu koja imaju jedinstvenu vrijednost za sve slogove. Relacije među podacima (tj. tablicama) se postavljaju preko sekundarnih ključeva (secondary key) koji su kopije primarnih ključeva iz tablica među kojim se definira relacija. Podržava referencijalni integritet podataka.

3. Objektne (ili objektno-orijentirane) baze podataka informacije pohranjuju kao objekte u objektno-orijentiranom programiranju. Objekt ima podatke i procese koji se primjenjuju na tim podacima.

– Objekt je definiran klasom. Postoji mogućnost nasljeđivanja.

– za multimedijalne aplikacije ili sustave sa kompleksnim podacima

4. Višedimenzionalne baze – MDB podataka (multidimensional database) su najnoviji tip bazi podataka koji je usmjeren olakšavanju dohvata podataka iz baze. Optimiziran za tzv. podatkovna spremišta (data warehouse) i tzv. aplikacije za online analitičko procesiranje.

46. Opišite hijerarhijske baze podatka, svojstva, prednosti i nedostatke.

– Hijerarhijske baze koriste hijerarhiju za definiranje relacija među podacima

– Podaci su organizirani hijerarhijski, u obliku stabla.

– Podržavaju relacije 1:1 i 1:M dok se M:N relacije teško ili nikako ne mogu prikazati hijerarhijskim relacijama podataka.

– Prednost im je brzo izvođenje pretraživanja.

– Svaki slog ima jedan roditeljski slog i više slogova djece

47. Opišite objektno-orijentirane baze podatka, svojstva, prednosti i nedostatke.

Objektne (ili objektno-orijentirane) baze podataka informacije pohranjuju kao objekte u objektno-orijentiranom programiranju. Objekt ima podatke i procese koji se primjenjuju na tim podacima.

– Objekt je definiran klasom. Postoji mogućnost nasljeđivanja.

– za multimedijalne aplikacije ili sustave sa kompleksnim podacima

48. Opišite višedimenzionalne baze podatka.

Višedimenzionalne baze – MDB podataka (multidimensional database) su najnoviji tip bazi podataka koji je usmjeren olakšavanju dohvata podataka iz baze. Optimiziran za tzv. podatkovna spremišta (data warehouse) i tzv. aplikacije za online analitičko procesiranje. Podatkovnim spremištima se nazivaju centralna spremišta svih važnih podataka poslovnog sustava. Mogu ga činiti različite baze podataka i datoteke koje zajednički čuvaju sve te podatke. Podaci se korisniku predstavljaju u obliku višedimenzionalnog niza, pri čemu je svaki podatak sadržan u jednoj ćeliji niza kojoj se može pristupiti preko višestrukih indeksa. Pogled na podatke je integriran direktno u strukturu podatka kroz jednu od dimenzija

49. Od kojih pet koraka se sastoji postupak prelaska sa logičkog na fizički model podataka?

1. pretvaranje entiteta u tablice ili datoteke
2. pretvaranje atributa entiteta u polja
3. dodavanje primarnih ključeva
4. dodavanje sekundarnih ključeva
5. dodavanje sistemskih podataka

50. Koje se tehnike koriste za optimizaciju brzine pristupa bazi podataka? Opišite ih.

– Denormalizacija – postupkom denormalizacije se namjerno unose redundancije u podatke koje ubrzavaju dohvat podataka

– Uskupljavanje (clustering) – reduciranje broja pristupa bazi fizičkim postavljanjem sličnih zapisa jednih blizu drugih (Intrafile i Interfile clustering)

– Indeksiranje – omogućavaju ga relacijske baze. To su dodatni podaci koji se spremaju za promatranu tablicu, ali višestruko ubrzavaju rad sa tablicom.