

Modulio “Informacinių sistemų pagrindai” egzamino klausimai (II dalis)
(Koliokviumo klausimai pateikti atskirame lape)

31. Reikalavimų nustatymo etapo turinys ir specifika (bendra charakteristika) (9pask, 3-4 psl.).

Tai mažiausiai apibrėžtas ir reglamentuotas IS kūrimo etapas.

To priežastys:

- tai mažiausiai techniškai ir labiausiai nuo konkrečios DS priklausomas procesas;
- poreikiai iš prigimties (dėl žmogiškojo faktoriaus) nėra apibrėžiami naudojant griežtus metodus, nes poreikių išskyrimas paprastai specifinis kiekvienam analizuojamam objektui

Reikalavimų nustatymo etapai:

- 1) Problemos apibrėžimas;
- 2) Galimybių analizė;
- 3) Reikalavimų įgijimas;
- 4) Reikalavimų analizė

32. Problemos apibrėžimas reikalavimų nustatymo metu (9pask, 5-8 psl.).

Pirmiausia pagrindžiamos sistemos kūrimo priežastys. Tradiciškai tai pateikiama techniniame pasiūlyme - sistemos kūrimo pagrindime. Tai griežtos formos neturintis dokumentas, kuriame apibrėžiama sistemos paskirtis ir jos kūrimo tikslai.

Kuriant sistemą gali būti siekiama bent vieno iš šių tikslų:

- 1) išspręsti problemą;
- 2) įvykdyti nurodymą;
- 3) išplėsti tam tikros veiklos galimybes;

Tai labai bendri tikslai, kurie turi būti detalizuoti, nurodant tiesioginę kuriamosios sistemos naudą ar reikalingumą organizacijai (įmonei).

Sistemos kūrimą pagrindžiantys faktoriai:

- 1) Informacijos apdorojimo galimybių išplėtimas (padidinti informacijos apdorojimo greitį, apimtį, paieškos galimybes, tikslumą bei operatyvumą);
- 2) Komunikavimo galimybių išplėtimas (organizacijos vidaus ir tarporganizacinių ryšių patobulinimas, padidinant pranešimų perdavimo greitį, tikslumą, sumažinant kainą ar padidinant informacinių srautų apimtį);
- 3) Valdymo patobulinimas, išvystant valdymo funkcijas, padidinant valdymo tikslumą, pilnumą, neprieštaringumą, patikimumą bei kontrolę;
- 4) Konkurencingumo padidinimas, patobulinant darbą su klientais, įgyjant pranašumo konkurentų atžvilgiu, optimizuojant tiekėjų ar realizavimo rinkos specialistų darbą bei operatyviai įsisavinant naujų gaminių gamybą;
- 5) Lėšų (finansų, kainų) valdymas, leidžiantis operatyviai ir tiksliai įvertinti finansavimo reikmes, faktines išlaidas žaliavoms, atlyginimams, energijos resursams, valdymui, detalizuojant išlaidas pagal organizacijos padalinius, darbuotojus, išlaidų straipsnius.

Pasiūlymą IS kurti (išvystyti) paruošia labiausiai patyrę IS kūrimo bei analizės specialistai. Tai turėtų trukti nuo kelių dienų iki kelių savaičių.

Pasiūlymo pagrindas - ataskaita, kuri gali turėti tokius punktus;

- sprendimo uždavinio apibrėžimas, jo svarbos paaiškinimas arba pagrindimas;

- siūlomo sprendimo metmenys, kur paaiškinama naujos IS vartotojams teikiama pagalba bei laukiama nauda (arba žalos išvengimas) ją sukūrus;
- reikalingos sistemos orientacinė apimtis.

Detalumo laipsnis priklauso nuo kuriamos sistemos apimties ir kūrimo etapų.

33. Sistemos kūrimo galimybių analizės paskirtis, sudėtis, rezultatai ir rezultatų panaudojimas (9pask, 9-10 psl., 20-23 psl.).

Jos metu įvertinamos pasiūlymo realizavimo galimybės, išanalizuojant uždavinį ir parenkant geriausią sprendimą DS kontekste.

Kartais gali būti pateikti keli alternatyvūs sprendimai, turintys skirtingą funkcijų diapazoną arba įgyvendinimo variantus.

Šio etapo metu suformuluojami aukšto lygio (apibendrinti, nedetalizuoti) funkciniai ir nefunkciniai reikalavimai.

Funkciniuose reikalavimuose numatomos sistemos atliekamos funkcijos.

Nefunkciniuose reikalavimuose pateikiami kokybiniai apribojimai sistemai, pavyzdžiui, maksimalus vartotojų skaičius, reikalavimai saugumui arba patikimumui.

Šio etapo metu turi būti atliktas sistemos kūrimo galimybių pagrindimas šiais aspektais:

- 1) sistemos atitikimas organizacijos funkcionavimo tikslams;
- 2) ekonominis kūrimo galimumas;
- 3) techninis kūrimo galimumas;
- 4) eksploatavimo galimumas.

Sistemos kūrimo galimybių analizės rezultatai

Šio etapo sudėtingumas:

- dauguma klausimų negali būti tiksliai atsakyti dėl nepakankamo DS žinojimo;
- be to vienas iš faktorių, lemiančių sistemos kūrimo sėkmę - IS projektuotojų komandas pasiruošimas darbui. Šis faktorius dažniausiai tiesiogiai neanalizuojamas.

Etapo metu, tyrimas atliekamas apytiksliai ir greitai, atmetant tokius pasiūlymus:

- neatitinkančius organizacijos vystymosi strategijos;
- techniniu požiūriu neįmanomus;
- duodama nauda nekompensuojančius išlaidų;
- vartotojui nesant pasirengusiam eksploatuoti sistemą.

Šį etapą vykdo analitikai, diskutuojantys su vartotojais (užsakovais) ir analizuojantys dokumentaciją bei kompiuterizuojamą objektą.

Paruošiama ataskaita, kurioje pateikiama:

- funkciniai (nedetalūs) reikalavimai;
- svarbiausi nefunkciniai reikalavimai;
- apžvelgiami visi keturi aukščiau minėti IS kūrimo galimumo aspektai.
-

Įvertinus finansines galimybes ir kitus poreikius, gali būti rekomenduojama:

- įsigyti jau sukurtą integruotą programų paketą, realizuojantį reikiamas funkcijas (pvz.: MS Dynamics AX, Navision, MFG/PRO, SAP, PeopleSoft, ...);
- kurti originalią sistemą.

Sistemai kurti gali būti numatyti du etapai:

- 1) kuriamas ir diegiamas "pilotinis" projektas, apimantis tik dalį sistemos funkcijų;
- 2) sistema plečiama iki pilno funkcionalumo.

Sistemos kūrimo galimybių analizės rezultatų panaudojimas projekto planavimui

Etapo metu surinkta informacija yra pagrindas IS kūrimo planui sudaryti.

Jis apima: finansavimo reikmes; laiko sąnaudas; projektuotojų reikmes.

Projekto planas dažniausiai yra hierarchinės struktūros, kuriame atskiri IS kūrimo etapai detalizuojami, numatant to etapo tikslus, reikalingus darbus, jų kokybės bei korektiškumo (savalaikiškumo) kontrolės būdus

Detaliam planavimui gali būti panaudotos diagramos, pateikiant tikslus darbų trukmės bei įvykdymo terminų įverčius, finansavimo reikmes bei darbuotojų resursus. Projekto papildomi dokumentai: techninė užduotis ir kalendorinis darbų grafikas.

34. Sistemos kūrimo galimybių analizė įvertinant organizacijos gyvavimo tikslus bei ekonominius faktorius (9pask, 11-14 psl.).

Sistemos atitikimo organizacijos funkcionavimo tikslams įvertinimas

Sistemos kūrimo idėja turi būti apibrėžta korektiškai ir remiantis objektyviomis priežastimis. IS kūrimo motyvu negali būti organizacijos padalinių tarpusavio nesutarimai, IS skyriaus prestižo padidėjimas ir pan.

Jei organizacija turi strateginį vystymosi planą, tuomet pasiūlymas kurti IS turi būti patikrintas atitikimo šiam planui aspektu

Šio etapo metu turi būti suformuluojami visos IS bei atskirų jos dalių kūrimo objektyvūs motyvai.

Ekonominis IS kūrimo galimumo įvertinimas

Pirmiausia turi būti įvertintos sistemos kūrimo finansinės išlaidos bei jos kūrimo alternatyvų duodama nauda.

Vienų uždavinių (funkcijų) kompiuterizavimas gali duoti daugiau naudos, o kitų - mažiau.

Be to, čia įvertinami ir nefunkciniai reikalavimai, pvz. atskirų uždavinių veikimas realaus laiko režimu, su fiksuotu vėlinimu ir pan. Tai tiesiogiai siejasi ir su techninėmis galimybėmis, tačiau ženkliai įtakoja išlaidos sistemai kurti.

Šalia to, būtina įvertinti tiesiogiai sistemos kūrimui reikalingus resursus:

- finansinės lėšos;
- žmonių ir laiko ištekliai.

Pirmiausia, tai:

- 1) atlyginimas sistemos kūrėjams: analitikams, projektuotojams, programuotojams, testuotojams;
- 2) vartotojo darbo laiko (lėšų) sąnaudos reikalavimams suformuluoti, apsimokymui, sistemos bandymams;
- 3) išlaidos techninei ir programinei įrangai

- Šis etapas nėra paprastas, nes sunku tiksliai numatyti sistemos duodamą efektą, kadangi jos kūrimo sėkmė priklauso nuo daugelio faktorių.
- Įvertinant IS kūrimo kainą, jos eksploatavimo kainos taip pat negalima pamiršti (kartais ji yra pakankamai aukšta dėl aptarnavimo kaštų, licencijavimo ir pan.)

35. Sistemos kūrimo galimybių analizė įvertinant techninius apribojimus ir eksploatavimo galimumą (9pask, 15-19 psl.).

- Įvertinama, ar IS kūrimo idėja suderinama su įmonėje egzistuojančia (naudojama) technologija.
- Jei taip yra, tuomet esama techninė bazė dažniausiai turi būti atnaujinta arba papildyta.
- Kai kurie sprendimai gali reikalauti naujausių technologijų (techninės ir programinės įrangos) įsisavinimo, kurias gali tekti integruoti su senosios IS ištekliais.

Reikia įvertinti nefunkcinių reikalavimų techninius aspektus:

- darbas realiame laike;

- reakcijos į vartotojo užklausas laikas;
- duomenų perdavimo (replikavimo) dažnis (apimtis) tarp atitolusių sistemos dalių (paskirstyta sistema, transakcijų mechanizmo apribojimai);
- duomenų bazės prognozuojama apimtis;
- maksimalus vartotojų skaičius.

Be to turi detalizuojami reikalavimai:

- sistemos išvystymo galimybėms (atvirumo aspektu);
- patikimumui;
- saugumui;
- sistemos DB-ės kopijavimui ir archyvavimui.

Kuriamos IS eksploatavimo galimumo įvertinimas

Būtina įvertinti organizacijos darbuotojų pasirengimą bei nusiteikimą eksploatuoti sistemą.

- Pavyzdžiui, darbuotojai gali būti patenkinti esama sistema ir nenorėti pasikeitimų (taip pasitaiko dėl netiesioginių faktorių). Dėl to bus neigiama reakcija diegiant sistemą.
- Taip įvyksta, jei sprendimas kurti sistemą priimamas aukščiausiu lygiu, sistemos kūrimo pasiūlymo ruošime nedalyvaujant jos tiesioginiams vartotojams.
- Reikia įvertinti ir organizacijos veiklos “tvarkingumo” aspektą (Netvarkos kompiuterizuoti praktiškai neįmanoma!)

Projekto sėkmės faktoriai:

- jo kūrimo ir diegimo palaikymas iš organizacijos vadovybės pusės;
- betarpiškas sistemos vartotojų dalyvavimas kuriant sistemą (poreikių nustatymas ir diegimas).

36. Reikalavimų įgijimo būdai ir jų charakteristika (9pask, 24-36 psl.).

Vartotojo reikalavimų įgijimo būdai:

- interviu;
- anketavimas;
- dalykinės srities (esamos sistemos) dokumentacijos analizė;
- kuriamos sistemos dokumentacijos analizė;
- stebėjimas.

Interviu - dažniausiai naudojamas reikalavimų įgijimo metodas.

leidžia vartotojui betarpiškai ir savarankiškai pareikšti pageidavimus sistemai; pirmuosius interviu vartotojas gali pradėti be didesnio pasirengimo

Etapai:

- susipažinti su bazine informacija;
- pagrįsti interviu tikslus (kokios informacijos reikia?);
- pasirinkti tinkamą DS specialistą;
- sudaryti interviu struktūrą (tvarkaraštį bei trukmę);
- numatyti klausimų pobūdį bei sudėtį.

Interviu struktūros:

piramidės ; piltuvėlio - deimanto –

Piramidės tipo interviu :

- pradedamas nuo specifinių klausimų;
- baigiamas apibendrinančiomis išvadomis;
- šis būdas naudojamas norint iš karto sukonzentruoti pokalbį apie tam tikrą klausimą, o naudojantis kitomis priemonėmis informacijos apie tai nepavyko gauti

Piltuvėlio tipo interviu :

- pradedamas nuo bendrų klausimų;
- išsivysto iki konkrečių klausimų detalizavimo;
- šis būdas suteikia interviu daugiau laisvės ir leidžia išryškinti daugiau detalių

Deimanto tipo interviu :

- pradedamas nuo specifinių klausimų;
 - aptiriamos apibendrinančios išvados;
 - baigiama specifiniais klausimais
 - šis būdas yra pirmųjų dviejų būdų kombinacija
-
- Interviu metu gauta informacija dažnai nėra vienareikšmiška;
 - pakankamai sunku užtikrinti gaunamos informacijos pilnumą

Anketavimas

Anketos naudojamos, kai reikia surinkti informaciją iš daugelio DS aktorių.

Naudotinas, kuomet :

- yra daug būsimosios sistemos vartotojų ir reikia įvertinti jų nuomonę;
- norint atlikti paruošiamąjį darbą prieš apklausą interviu būdu (kad atkreipti dėmesį, sukoncentruoti mintis, apmąstyti iš anksto)

Anketavimo naudojimo sunkumai:

- vartotojui sunkiau atsakyti į atviro pobūdžio klausimus;
- tai ilgiau užtrunka (analitikui) nei interviu;
- anketoje gali būti nenumatyti svarbūs klausimai, kurie išryškėja interviu metu ir gali būti operatyviai aptarti;
- anketą reikia sudaryti, išplatinti ir išanalizuoti

Interviu, anketos atžvilgiu, geresnis dėl betarpiškumo ir leidžia atskleisti prieštaraivimus bei konfliktus anksčiau pateiktoje medžiagoje.

Tačiau interviu metu viską reikia ne tik užfiksuoti, bet ir suprasti

Dalykinės srities (esamos sistemos) dokumentacijos analizė

Tai, kartu su interviu, vienas svarbiausių reikalavimų įgijimo šaltinių.

Analizė atliekama tiriant esamoje sistemoje naudojamus duomenis (informaciją): kiekybinio ir kokybinio pobūdžio

Kiekybiniai duomenys: pirminiai dokumentai, ataskaitų duomenys, veiklos instrukcijos.

Kokybiniai duomenys (siejasi su organizacijos darbo kultūra): reklamos dokumentai, skelbimų lentos pranešimai, atmintinės ir pan.

Pavyzdžiui, organizacija, kurioje yra daug dokumentų, pasirašytų vadovaujančio personalo ir reglamentuojančių veiklos taisykles ar pateikiančių įspėjimus, yra daugiau centralizuota.

Organizacija, kurioje paplitę vieši pranešimai skelbimų lentoje remiasi daugiau decentralizuotu valdymo modeliu.

Tokie pastebėjimai leidžia nuspręsti, ar IS funkcionavimo rezultatai turi būti orientuojami į vadovus, ar žemesniam personalui.

Kuriamos sistemos dokumentacijos analizė

Tai pastabos ir kiti dokumentai apie vartotojo poreikius bei apribojimus kuriamai sistemai, pateiktos ankstesnių naujosios sistemos kūrimo žingsnių metu.

Tai daugiausia bendri reikalavimai, pateikti:

- sistemos kūrimo techniniame pasiūlyme;
- problemos apibrėžimo ataskaitoje;
- sistemos kūrimo galimumo apibrėžime.

Stebėjimas

Šis metodas naudojamas, kai:

- kiti šaltiniai neduoda rezultatų arba jų nėra;
- kiti šaltiniai duoda prieštarigus rezultatus ir reikia priimti vieningus sprendimus;

Stebėjimas gali būti atliekamas dviem būdais:

- betarpiškai stebint vartotoją ir netgi pateikiant klausimus apie atliekamą darbą (integruotas su interviu metodas);
- stebėtojai tiesiogiai nedalyvaujant, o naudojant technines stebėjimo priemones

Poreikių įgijimo metu nustatomi didesnė dalis vartotojo reikmių, tačiau vėliau jie dar turi būti detalizuojami ir patvirtinami

37. Reikalavimų nustatymo etapo sunkumai (9pask, 38-45 psl.).

I. Produktyvumo (darbo našumo) problemos

- Kyla dėl klaidingos prielaidos, kad “reikalavimai - tai objektyvus faktų rinkinys”; “jie gerai žinomi nuo IS kūrimo pradžios”, “yra fiksuoti ir nekinta”
- Kita nekorektiška prielaida, kad “projektavimo resursai gali būti įvertinti iš anksto pakankamai tiksliai”.

Deja, šios prielaidos nėra teisingos !

Poreikiai kinta dėl įvairių priežasčių

Poreikių kitimo priežastys:

I.1. Poreikiai nestabilūs ir vartotojai keičia savo nuomonę:

- a) įsigilinus į problemą (naujas požiūris);
- b) perpratus informacinių technologijų teikiamas galimybes;
- c) plečiant sistemos galimybes ir dėl to keičiantis vartotojo poreikiams;
- d) dėl vartotojų tarpusavio prieštaravimų (vyraujanti nuomonė kartais pasikeičia likvidavus prieštaravimus);

- e) dėl didelio suinteresuotų DS specialistų skaičiaus

I.2. Išorės faktoriai gali įtakoti pokyčius (IS kuriama netrumpą laiką - iki 2 ir daugiau metų) :

- a) pasikeičia įstatymai ;
- b) pasikeičia organizacijų struktūra (susijungia, pasidalina);
- c) pakinta organizacijos veikla;
- d) atsiranda modernesnė techninė bei programinė įranga.

I.3. Sistemos diegimas, neįvedus pakeitimų, gali būti neįmanomas:

- a) diegimo metu paaiškėja, kad tam tikri nefunkciniai reikalavimai netenkinami (pavyzdžiui, sistemos reakcijos laikas);
- b) diegiant atsiskleidžia trūkumai funkcinuose reikalavimuose

I.4. Komplikuotas ir nepakankamas projekto valdymas:

- a) didelės paklaidos įvertinant projektavimo resursus (laiką ir lėšas);
- b) nepatyręs personalas.

Manoma, kad projektavimo resursai, įskaitant projektavimo kainą, programinės ir techninės įrangos įsigijimo kainą, projektuotojų kiekį ir kvalifikaciją, projektavimo trukmę, gali būti nustatyti projektavimo pradžioje, sistemos kūrimo galimybių nustatymo fazėje.

Tačiau tam tikslui yra tik labai apytikslės metodikos, neužtikrinančios pakankamo tikslumo ir neteikiančios garantijų.

Projekto valdymas žymia dalimi remiasi projektavimo personalo patirtimi ir intuicija.

Kadangi užsakovai tradiciškai siekia trumpinti projektavimo trukmę ir mažinti kainą, resursų įvertinimas dažnai būna pernelyg optimistinis (užsakovo aspektu).

Todėl projektavimo eigoje resursus tenka tikslinti (laiką ir lėšas)

Poreikių nustatymas yra iteratyvus procesas, todėl projektui reikalingi resursai gali išaugti, o tai apsunkina projekto valdymą

II. Kokybės problemos

Kokybė gali būti žema dėl to, kad;

- a) analizės metu taikytos metodikos neįvertino specifinių žinių apie DS, kurios galėjo pagelbėti nustatant poreikius. To dažniau išvengia patyrę sistemos analitikai;
- b) analizei naudojami modeliai nėra išvystyti, naudojamos tik analitiko intuicija, nesuteikiant jam paramos specialių priemonių pagrindu.
Kuo didesnė sistema, tuo daugiau šansų, kad bus klaidų poreikių specifikacijoje;
- c) kokybė krenta dėl pokyčių reikalavimuose informacijos sistemai;
- d) nepatyrusi projektuotojų komanda.

Poreikių nustatymas yra iteratyvus procesas, todėl projektui reikalingi resursai gali išaugti, o tai apsunkina projekto valdymą

38. Reikalavimų nustatymo etapo sunkumų sprendimo būdai (9pask, 46-50 psl.).

1. Patvirtinimas ir prototipo sudarymas (labiausiai tinka I.1 a) b) problemoms spręsti)
 - Tai metodai, kurie vartotojui suteikia galimybę patvirtinti poreikių analizės rezultatus;
 - Jie ne tik leidžia aptikti klaidas, padarytas analitiko, bet ir užtikrina iteracinę poreikių nustatymo technologiją;
 - šios priemonės tik patvirtina, kad poreikiai dar nėra pilnai žinomi projektavimo pradžioje
2. Grupiniai pasitarimai (labiausiai tinka I.1 d) - e) problemoms spręsti)
 - jų metu suinteresuoti vartotojai sukviečiami į pasitarimus, kurių metu nesutarimai reikalavimuose detalizuojami, analizuojami ir išsprendžiami;
 - kartais organizuojama pasitarimų serija;
 - gali būti organizuojami atskiri pasitarimai vadovams ir kitų lygių personalui.
3. Atgalinė inžinerija (labiausiai tinka II a) problemai spręsti)
 - metodas naudojamas kai egzistuoja sena sistema ir reikia ją transformuoti į naują sistemą
 - metodas naudoja CASE priemones, kad apdoroti esamos sistemos dokumentaciją, dažniausiai pateikiamą programų tekstų bei DB struktūrų fizinių specifikacijų forma;
 - rezultatas - suformuotas apibendrintas esamos sistemos aprašas;
 - jo pagrindu, dalis senosios IS sprendimų perkeliama į naująją sistemą
4. Šiuolaikinių reikalavimų analizės metodų bei priemonių panaudojimas (II b) problemai spręsti)
 - panaudojami naujausi struktūrinės arba objektinės analizės metodai, leidžiantys suklasifikuoti ir struktūrizuoti neformalų reikalavimų aprašymą;
 - panaudojamos reikalavimų specifikavimo šiuolaikinės priemonės: RequisitePro, Doors ir kt.
5. Projekto valdymo patobulinimas (I 4) problemai spręsti)
 - naudotini projekto valdymą automatizuojantys paketai, nors jie neturi priemonių pateikiant tikslesnį projekto resursų įvertinimą (pvz. MS Project);
 - daugiau dėmesio būtina skirti projekto valdymui organizuoti (dalis informatikų tinka individualiam darbui, o ne vadovavimui-organizaciniam darbui);
 - naudotinas ne optimistinis, o realistinis resursų vertinimas (patirties pagrindu)

39. Reikalavimų analizės sudėtis ir nauda. Reikalavimų nustatymo etapo būdingos savybės (9pask, 37 psl., 51 psl.).

Jos metu neformalūs reikalavimai transformuojami sudarant struktūrizuotą jų aprašymą - reikalavimų specifikaciją.

Nauda:

- analitikui paprasčiau surasti prieštaravimus;
- lengviau pastebėti informacijos trūkumą;

- tvarkingą specifikaciją galima pateikti vartotojui patikrinti (patvirtinti)
- Reikalavimų specifikacijai sudaryti gali būti naudojamas tam tikras modelis, užtikrinantis adekvatų reikalavimų atvaizdavimą.

Poreikių įgijimas - iteratyvus procesas;

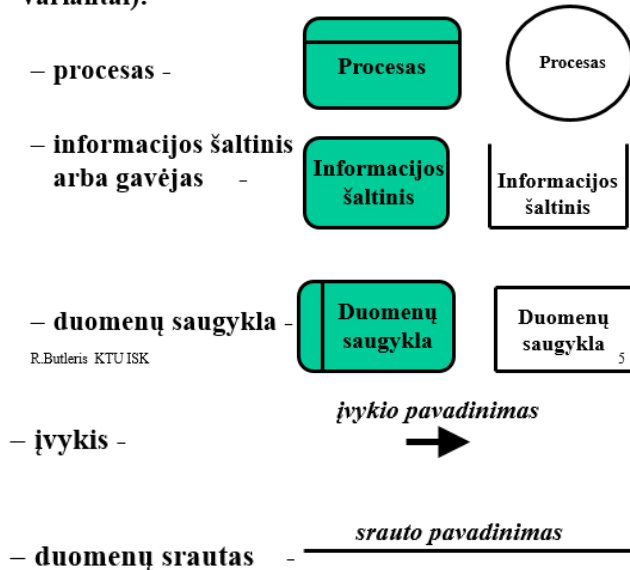
- Pagrindinis sunkumas - reikalavimų analizės fazė neturi pilnai išvystytų metodų, kurie tarnautų kaip išbaigtos reikalavimų specifikacijos sudarymo priemonė (priežastis - reikalavimus pateikia žmogus, kuris mąsto pakankamai neformaliai ir neišbaigta);
- Pradinėse analizės fazėse sunku užtikrinti reikalavimų specifikacijos pilnumo bei neprieštaringumo reikalavimus;
- Reikalavimų nustatymo fazė lemia visos IS kūrimo sėkmei;
- Jei klaidos ištaisymas reikalavimų nustatymo fazėje kainuoja 1 Lt, tai ta pati klaida, išaiškinta eksploataavimo ir priežiūros fazėje gali kainuoti 1000 Lt ir daugiau

40. Duomenų srautų diagrama (8pask).

- Procesų dekompozicijos diagrama detalizuoja procesus, tačiau dažnai reikalinga atvaizduoti procesų sąryšius per jų įėjimus ir išėjimus;
- Šiam tikslui gali būti panaudota Duomenų srautų diagrama (Data Flow Diagram), kuri suformuoja labiau integruotą sistemos atvaizdą;
- DSD tinka labiau, kuomet duomenys apdorojami daugelio procesų

DSD sudėtis

- DSD sudaromos iš 5 tipų simbolių (yra keli žymėjimų variantai):

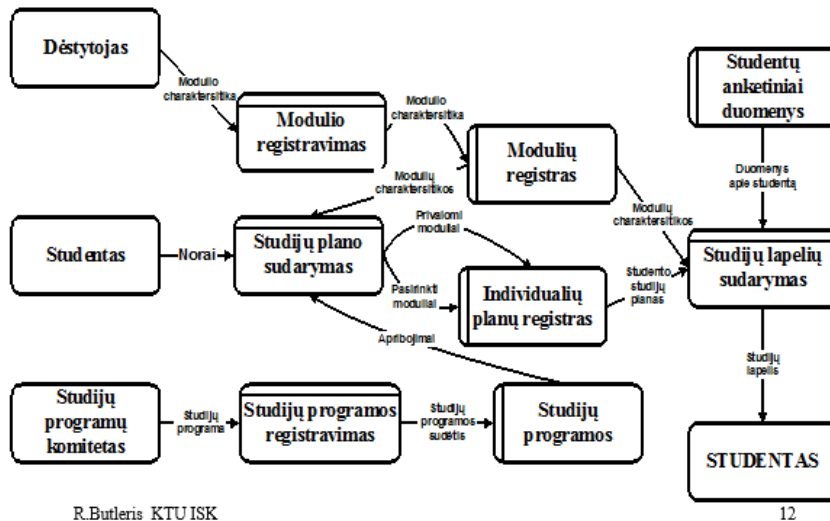


DSD sudarymo taisyklės

- 1) kiekvienam procesui suteikiamas unikalus vardas (kartais šalia vardo suteikiamas numeris, nusakantis proceso lygį procesų dekompozicijos diagramoje; pavyzdžiui, 2.1; 3.4.2);
- 2) Visi procesai privalo turėti įėjimą ir išėjimą, kurie atvaizduojami duomenų srauto žymėjimu (lanku), jungiančiu procesą su kitais diagramos elementais;
- 3) Duomenų srautas taip pat privalo turėti unikalią vardą, kuris užrašomas virš lanko;
- 4) Procesų vyksmą sukeliantys įvykiai, kurie neįeina į sistemą (yra išoriniai), parodomi šalia atitinkamo proceso trumpomis strėlėmis su užrašytu įvykio vardu
- 5) Į sistemą neįeinančios esybės vaizduoja išorinius informacijos šaltinius arba gavėjus; diagramoje nepateikiama informacija apie jų elgesį bei struktūrą, o įvardinami tik iš jų ateinantys arba į juos išeinantys duomenų srautai;

- 6) Duomenų saugyklos tarnauja kaip “buferis” duomenims saugoti tuo metu, kai vienas procesas baigiasi, o kiti procesai tik po tam tikro laiko naudojami jo vyksmo rezultatais - išėjimo duomenimis;
- 7) Duomenų saugyklos turi unikalų vardą; duomenų saugyklos jungti tarpusavyje betarpiškais ryšiais negalima;
- 8) Duomenų saugyklos tarnauja kaip procesų įėjimo duomenų šaltinis arba išėjimo duomenų saugykla.

DSD pavyzdys



12

- Duomenų srautų diagramos nėra visapusiškai tinkama priemonė analizės etapui, kadangi remiasi duomenų srauto ir duomenų saugyklos sąvokomis, t.y., akivaizdi orientacija išimtinai į duomenų srautų modeliavimą;
- Labiau tinka toms DS, kuriose figūruoja aptarnavimo procesai, pavyzdžiui, bankų ar draudimo bendrovių veikla, kuriose dominuoja informaciniai srautai;
- Gamybinio pobūdžio veiklai modeliuoti DSD tiesiogiai sunkiau pritaikoma, kadangi čia dominuoja žaliavų ar produkcijos srautai;
- Tačiau yra galimybė vietoje duomenų srautų, atvaizduoti produkcijos arba žaliavų srautus, gaminių arba žaliavų saugyklas bei jų apdorojimo procesus;
- Jei sudaromas ER modelis, tuomet materialius srautus bei saugyklas atvaizduojančios DSD elementai galėtų būti panaudoti ER modelio komponentėms - esybėms išskirti;
- Negalima pamiršti, kad bet kuris materialus srautas gali būti valdomas tik tuomet, kai šalia jo egzistuoja ir informacinis srautas!
- DSD per procesus siejasi su procesų dekompozicijos (funkcijų hierarchijos) diagrama;
- kiekvienas procesų dekompozicijos diagramos hierarchijos lygis gali turėti atitinkamo lygio DSD;
- DSD naudojami duomenų srautai ir saugyklos gali pilnai ar dalinai sutapti su ER modelyje naudojamomis komponentėmis;
- yra galimybė atlikti verifikavimą, tikrinant DSD, FHD ir ERD integralumą;
- praktiškai sudaroma kelių lygių DSD, dekomponuojant procesus ir pateikiant jų vidinę informacinę architektūrą;
- pagrindinė DSD paskirtis - atspindėti atitinkamo detalumo lygio procesus bei jų tarpusavio sąryšius; esant poreikiui, iširti jų sudėtį ir atlikti detalią analizę;
- galima sudaryti analizės lygio ir projektavimo lygio DSD. Analizės lygio DSD atspindi DS esamus duomenų srautus, o projektavimo lygio DSD - kuriamos IS duomenų srautus;

- sudarant analizės lygio DSD būtina išvengti DB failų, kompiuterinių ataskaitų ir kitų IS sudėtinių dalių modeliavimo;
- projektavimo lygio DSD atspindi kompiuterizuotų procesų, duomenų saugyklų bei sistemos vartotojų tarpusavio informacinius mainus;
- projektavimo lygio DSD gali būti panaudota vartotojo ir kompiuterio interfeisui modeliuoti (geriau tinka būsenų kaitos diagrama);
- DSD trūkumas - nėra išvystytų priemonių klaidoms ir išimtinėms situacijoms atvaizduoti;
- Nėra galimybių atvaizduoti procesų valdymo struktūrą;
- Modeliuojant realias DS kai kurie DSD atvejai yra pakankamai painūs ir sudėtingi

41. Būsenų kaitos diagrama (8pask).

Paskirtis - atvaizduoti sistemos būsenas bei jų kitimo sąlygas

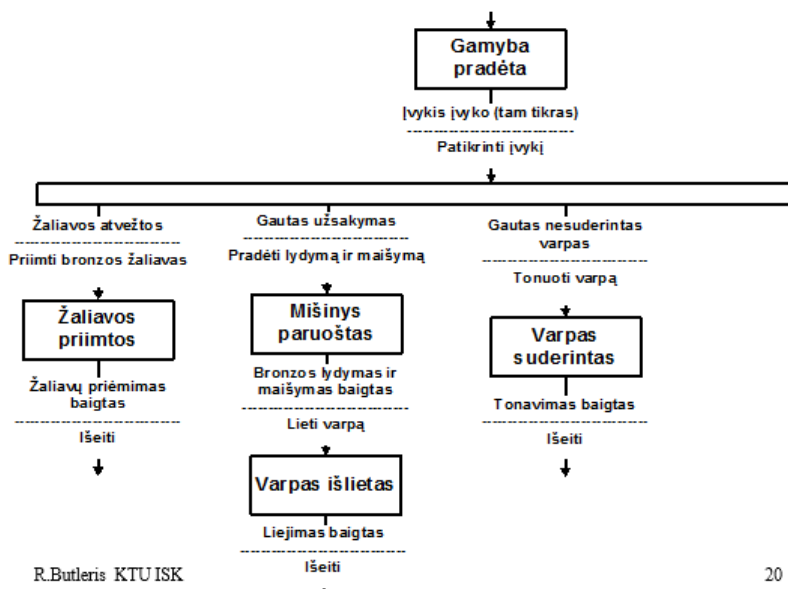
- dažnai naudojama specifikuojant vartotojo ir sistemos sąsają (vartotojas įveda komandas ir duomenis, o kompiuteris pateikia atsakymus, atitinkamai reaguoją);

BKD naudinga tuomet, kai analizuojama sistema pasižymi diskretiškų, lengvai išskiriamų būsenų aibe; kai svarbu atvaizduoti būsenų kaitos eiliškumą bei sąlygas; kai būsenų kaitos detalės (smulkmenos) nėra labai svarbios, t.y., nėra sudėtingo, kompleksinių sąlygų įtakojama būsenų kitimo.

BKD sudaroma iš trijų elementų:

- būsenos, atvaizduojamos stačiakampiu, su įrašytu būsenos pavadinimu (tuščiu stačiakampiu vaizduojamos būsenos, iš kurių, priklausomai nuo susiklosčiusių sąlygų, gali būti pereinama į skirtingas galines būsenas);
- būsenos kaitos žymėjimo, atvaizduojamo lanku, jungiančiu išeities (pradinę) būseną su paskirties (galine) būsena;
- sąlygos/veiksmo žymės, pateikiamos šalia atitinkamo lanko, sąlygą atskiriant nuo veiksmo horizontalia arba pasvirusia linija; virš linijos užrašoma būsenos perėjimo (kaitos) sąlyga, o žemiau linijos - veiksmai, atliekami kintant būsena.

Būsenų kaitos diagramos pavyzdys (pradžia)



R.Butleris KTU ISK

20

- DSD ir BKD tiesiogiai siejasi, kadangi DSD įvykiai atitinka BKD sąlygas, o BKD veiksmai atitinka DSD procesų turinį;
- BKD būsenos atspindi momentus tarp DSD procesų vyksmo, kai vienas procesas baigiasi, o kitas dar nėra prasidėjęs; procesų darbo rezultatai tuomet saugomi saugyklose;
- BKD veiksmai atitinka procesų dekompozicijos diagramoje vaizduojamų procesų sudėtį.

- naudojant BKD vartotojo ir sistemos sąsajai atvaizduoti, įvykiams gali atitikti tam tikros komandos įvedimas arba ikonos (meniu punkto) išrinkimas;
- sunkumai kyla parenkant BKD būsenų vardus; yra galimybė netiksliai išreikšti sąlygas arba veiksmus, o tuomet sunku atsekti ryšį tarp BKD ir DSD bei FHD

42. Srauto diagrama, struktūrizuota kalba ir veiksmų diagrama (8pask, 25-34 psl.).

•naudojama atvaizduoti procesus bei jų valdymo struktūrą detalesniame lygmenyje;

•dažnai pritaikoma parodyti procesų įėjimo ir išėjimo duomenų srautus

Naudojami simboliai:

•srauto pradžia ir pabaiga -



•valdymo srauto kryptis -



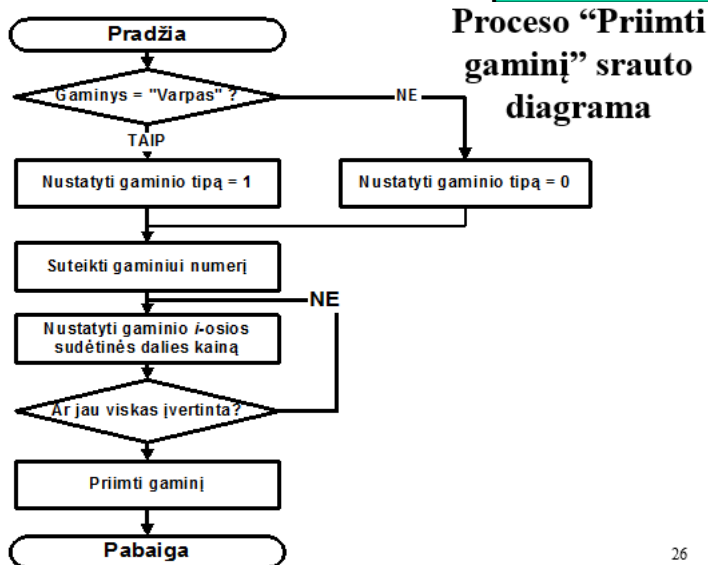
•alternatyvos - pasirinkimo blokas -



•procesas -



R. Butleris - KTU ISK



26

- tai tradicinė priemonė programoms projektuoti, leidžianti atvaizduoti sudėtingą valdymo struktūrą;
- sudėtingiems procesams atvaizduoti ši priemonė nėra labai patogi, nes schema persipina įvairiais procesų tarpusavio ryšiais ir tampa sunkiau suprantama.

Struktūrizuota kalba (*Structured English*)

Tai ribotos sintaksinės struktūros tekstinis procesų aprašymas žemiausiu detalumo lygiu

Yra įvairių struktūrizuotos kalbos variantų, kurių kiekvienas naudoja apribotą ir griežtai apibrėžtą sintaksinę struktūrą. Pavyzdžiui, apsiribojama:

- fiksuotais veiksmažodžiais: "papildyti -add"; "perkelti - move"; ...
- loginiais operatoriais: "mažiau nei... - less than"; ...
- loginėmis išraiškomis: "Kol <sąlyga> - atlikti ..., kitu atveju - ..." - "While ... Do ... Else ..."
- leidžia sumažinti tradicinės kalbos dviprasmiškumą;
- aprašo procesų apdorojimo operacijas, jų valdymo struktūrą bei duomenų srautus;
- nesudėtinga modeliuoti pasirinkimo veiksmus bei iteracijas;

- labiausiai pasitarnauja kaip detalus programų logikos aprašymas, kurio pagrindu galima atlikti programinę realizaciją

Struktūrizuotos kalbos trūkumas - sunkiau suprantamas aprašas esant didesniai sprendimo priėmimo arba nuoseklių veiksmų skaičiui

Proceso "Priimti gaminį" struktūrizuotas aprašas

BEGIN

Gaminio perdavimas

IF Gaminys = "Varpas" **THEN**

Nustatyti gaminio tipą = 1

ELSE

Nustatyti gaminio tipą = 0

Suteikti gaminiiui numerį

WHILE Yra neįvertintų gaminio sudėtinių dalių

DO

Nustatyti *i*-osios sudėtinės dalies kainą

Priimti gaminį

END

R. Butleris KTU ISK

30

PRADŽIA

Gaminio perdavimas

JEI Gaminys = "Varpas" **TAI**

Nustatyti gaminio tipą = 1

KITU ATVEJU

Nustatyti gaminio tipą = 0

Suteikti gaminiiui numerį

KOL Yra neįvertintų gaminio sudėtinių dalių

ATLIKTI

Nustatyti *i*-osios sudėtinės dalies kainą

Priimti gaminį

PABAIGA

R. Butleris KTU ISK

31

Veiksmų diagrama (*Action diagram*)

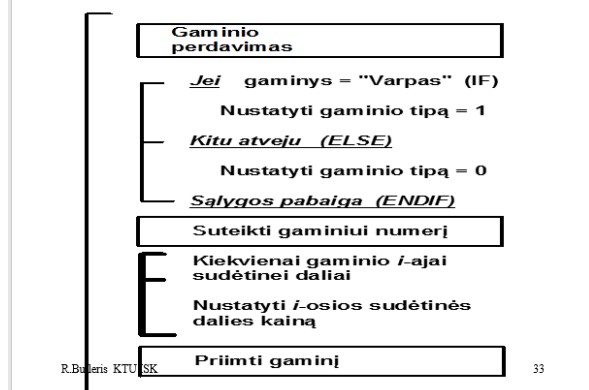
Tai integruotas tekstinio ir grafinio pobūdžio proceso aprašymas;

Pagrindinis privalumas - galimybė grafiškai atvaizduoti hierarchines priklausomybes tarp procesą sudarančių veiksmų bei jų grupių

Naudojami žymėjimai:

- tarpusavyje susietiems veiksmams vaizduoti naudojami skliaustai;
- susietų veiksmų grupės įjungiamos į kitas grupes naudojant atitraukimą;
- iteraciniai veiksmai žymimi pastorintais skliaustais, viršuje pateikiant sudvigubintą liniją

Proceso "Priimti gaminį" veiksmų diagramos pavyzdys



R. Butleris KTU ISK

33

- veiksmų diagrama naudojama žemiausiam proceso detalumo lygiui atvaizduoti, prieš pereinant prie kodavimo (programavimo) fazės;
- esant sudėtingai proceso sudėčiai, diagrama įgyja sudėtingą hierarchinę struktūrą, nors yra pakankamai vizuali ir gali būti panaudojama CASE priemonėse

43. Naši-Šneidermano struktūrograma (8pask).

- skirta vizualizuoti proceso valdymo struktūrą
- panaši į veiksmų diagramą

Naudojami simboliai:



- struktūrograma kuriama remiantis principu “iš visaus žemyn”, pradedant nuo išorinio ciklo

35

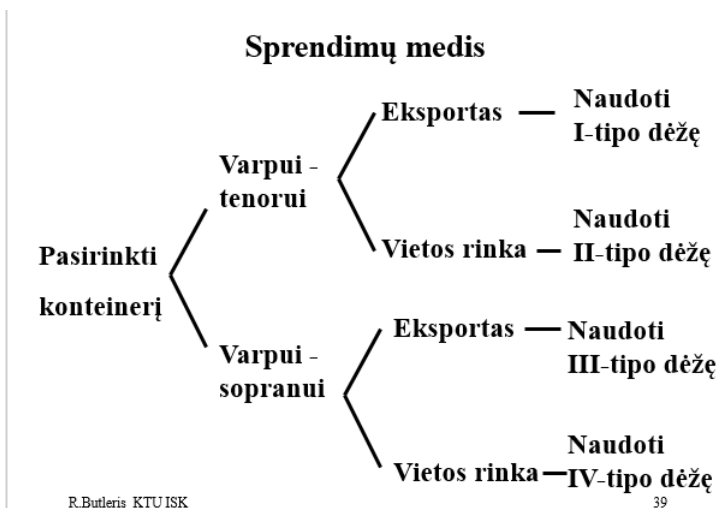


- nėra tokių vizualių galimybių atvaizduoti tarpusavyje susietų veiksmų poaibius, kaip veiksmų diagramoje;
- nėra taip paprasta atlikti modifikavimus, kaip veiksmų diagramoje;
- pasirinkimo ženklas gali būti skirtingo dydžio, priklausomai nuo pasirenkamų alternatyvų apimties (veiksmų kiekio vienu ir kitu atveju)

44. Sprendimų medis ir sprendimų lentelė (8pask).

Naudojamas aprašyti procesą arba proceso fragmentą, kuriame yra alternatyvių veiksmų

- Medis susideda iš aibės viršūnių, atitinkančių sąlygas, ir aibės lankų; viršūnės gali būti detalizuojamos, o žemiausiame lygmenyje esančios terminalinės viršūnės atitinka veiksmus;
- Medis dažniausiai vaizduojamas iš kairės į dešinę, tačiau gali būti pateikiamas ir vertikaliai;
- Naudojamas, kai yra didelis skaičius sąlygų, apibrėžiančių veiksmų vykdymo galimybes



Sprendimų lentelė

Varpas – tenoras	T	T	F	F
Eksportas	T	F	T	F
Naudoti I-tipo dėžę	X			
Naudoti II-tipo dėžę		X		
Naudoti III-tipo dėžę			X	
Naudoti IV-tipo dėžę				X

45. Esysbės gyvavimo istorijos diagrama (8pask).

- Skirta atvaizduoti su konkrečia esybe susijusius procesus bei jų valdymo struktūrą;
- Tai tarpinė atvaizdavimo priemonė, lyginant struktūrizuotą kalbą ir procesų dekompozicijos diagramą;
- Diagramoje parodomi tik esybės būsenų kaitos procesai;
- Visi su procesais susiję objektai turi būti esybės modelio komponentai (atributai, ryšiai arba pati esybė)
- procesai gali būti panaudojami tik su tam tikrais esybės egzemplioriais, o ne su visais egzemplioriais (taigi, įvertinami tam tikri apribojimai)

Pagrindiniai konstrukciniai elementai:

1. Seka. Diagrama sudaroma pradedant esybės atsiradimą sąlygojančiais procesais ir baigiant esybės gyvavimo ciklo pabaigą. Tai atspindi tam tikrą diagramos blokų pateikimo eiliškumą - sutvarkymu iš kairės į dešinę.

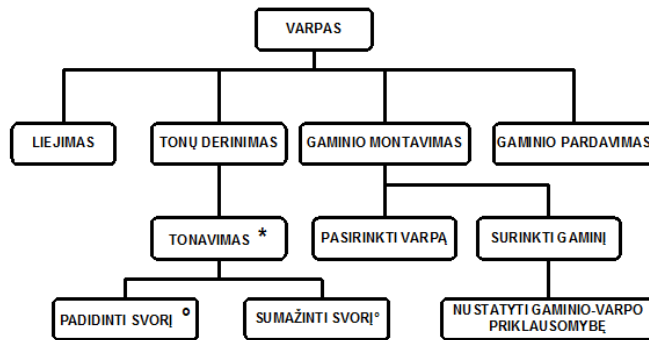
Procesai žymimi stačiakampiais su įrašytu proceso pavadinimu

2. Iteracijos. Tai parodoma žvaigždute (*) šalia proceso pavadinimo. Tai iliustruoja, kad procesas gali vykti nuo nulio iki tam tikro skaičiaus kartų.

3. Pasirinkimas. Tai parodoma rutuliuku (°) šalia proceso pavadinimo. Tai iliustruoja alternatyvius procesus, kurie gali vykti vienas arba kitas, bet ne abu kartu. Diagramoje neparodomos pasirinkimo sąlygos.

4. Diagramos viršuje pateikiamas blokas iliustruoja esybę, o žemiau esantys blokai - su ja susijusius procesus

Esybės gyvavimo istorijos (ciklo) diagrama



Privalumai:

- leidžia susieti procesus su esybėmis ir parodo, kurios esybės yra perteklinės - neturi susijusių procesų arba leidžia išskirti naujas esybes, nes atskleidžiamas procesų egzistavimas, kuriems reikalingos esybės nėra ER modelyje
- leidžia atlikti procesų valdymo kontrolę tarpiniame detalumo lygmenyje

Trūkumas: dėl sintaksinių apribojimų kartais tenka įvesti perteklinius blokus (tonavimas ir tonų derinimas)

46. Taisyklės samprata (10pask, 4-7 psl.).

- Taisyklės sąvoka organizacijoje turi daug įvairių interpretacijų. Pavyzdžiui, gali būti taisyklės:
 - nusakančios rūkymo patalpose tvarką;
 - nurodančios gamybinių ataskaitų pateikimo vadovybei dažnumą;
 - ribojančios priimamų darbuotojų amžių;
- Nagrinėsime tas taisykles, kurios vienaip ar kitaip įtakoja ar riboja tam tikrus DS objektus, kurie atvaizduoti ER modelyje
- Taisyklės iš tikrųjų galima apibrėžti plačiau, nei DS specifikacijos struktūrinės komponentės apribojimas
- Struktūrinė specifikacijos komponentė apima tik nedidelę dalį vartotojo reikalavimų, susijusių su egzistuojančiomis esybėmis, ryšiais ir atributais. Tačiau vartotojo reikalavimuose yra daug taisyklių, kurias reikia užregistruoti. Tuo tikslu sudaromas taisyklių modelis, kuriame taisyklės sukaupiamos deklaratyvioje formoje (ne procedūrinėje)
- Istoriskai, taisyklių sąvoka kilusi iš DB pilnumo apribojimų ir išvystyta konceptualaus modeliavimo srityje
- Viena iš priežasčių, dėl ko reikia modeliuoti taisykles, yra tai, kad struktūrinė DS specifikacijos komponentė remiasi objektų aibėmis, o ne atskirais jų egzemplioriais. Joje neįmanoma užregistruoti jokių poreikių, susijusių su individualiais egzemplioriais;
- Kita priežastis - poreikis sudaryti nepriklausomą taisyklių saugyklą (Business rules repository), kurioje taisyklės būtų saugomos autonomiškai nuo programinio kodo, taip užtikrinant didelį sistemų lankstumą bei galimybę operatyviai adaptuoti sistemų funkcionalumą pasikeitus sąlygoms (nekeičiant programinio kodo)
- Taisyklės gali būti įvairiai išreiškiamos, pavyzdžiui, natūralia kalba. Egzistuoja taisyklių modeliavimo metodikos:
 - GUIDE projektas,
 - Roso grafiniai taisyklių modeliai
- Taisyklių modelyje gali būti naudojamos tikslesnės nei natūrali kalba priemonės:
 - aibių specifikacijos (domeno lygio apribojimai)
 - loginiai apribojimai.
- Be to, taisyklės specifikuojamos remiantis struktūrinės modelio komponentės sąvokomis

- **Taisyklės apibrėžimas (10pask, 8-11 psl.).**

1. Taisyklės, tai yra teiginiai apie DS būsenas; jos nusako galimas sistemos būsenas;

2. Sistema čia suprantama kaip objektų (esybių, atributų ir ryšių) egzempliorius, apibrėžtus struktūrinėje dalykinės srities specifikacijos dalyje.

Pavyzdžiui, jei struktūrinė konceptualaus modelio dedamoji sudaryta iš esybių “Prekė”, “Pirkėjas” bei ryšio “Pirkėjas - Prekė”, tuomet “sistema” (dažnai ji gali būti suprantama kaip DB) galėtų būti sudaryta iš tokių tipų egzempliorių:

PIRKĖJAS	PREKĖ	PIRKĖJAS-PREKĖ
P1	S1	P1 - S1
P2	S2	P2 - S2
P2	S3	P3 – S3

3. Sistemos būseną tam tikru laiko momentu nusakoma šių komponentių:

- objektų egzempliorių reikšmių, pavyzdžiui, aukščiau pateikti trys esybės “Pirkėjas - Prekė” egzemplioriai;
- objektų savybių aibės, pavyzdžiui, pirkėjų egzempliorių skaičius arba prekės egzempliorių vidutinis skaičius, tenkantis pirkėjo egzemplioriams

4. Taisyklės gali būti teiginiai, nusakantys apribojimus:

- a) struktūrinės komponentės objektams;
- b) tam tikriems objektų egzemplioriams;
- c) nustatyti objektų savybes arba egzempliorių aibių savybes

Pavyzdžiai (atitinka apribojimų klasifikaciją pagal punktų žymėjimą):

- a) visi prekių egzemplioriai turi būti susieti su pirkėjais (pirkėjo esybe);
- b) prekę S1 gali pirkti tik pirkėjas P1 (specifiniai apribojimai);
- c) pirkėjų skaičius negali viršyti 1000 (nustato esybės “pirkėjas” savybę)

47. Taisyklių modeliavimo sunkumai (10pask, 12-19 psl.).

- Taisyklės tradiciškai išreiškiamos procedūrinėje formoje (taikomosiose programose, duomenų reikšmių redagavimo taisyklėse). Jų išraiškos yra žemame abstrakcijos lygyje.

Pavyzdžiui:

1. Procedūrinė išraiškos forma:

Nagr. taisyklę, apribojančią esybės “Pirkėjas” egzempliorių aibę iki 1000.

Deklaratyviai tai galima užrašyti taip:

KIEKIS (pirkėjai) <= 1000

Procedūriškai tai galima užrašyti taip:

IF Kiekis (pirkėjai) < 1000 THEN

 Įvesti_naują_tiekėją

ELSE

 WRITE (“Papildymas negalimas. Leistinas pirkėjų kiekis viršytas”)

Procedūrinėje formoje pateikti trys specifikacijos elementai:

- pati taisyklė sąlyginėje dalyje;
- atliekami veiksmai, kai taisyklė tenkinama;
- reakcija, kai taisyklė netenkinama

Tokia taisyklės forma (procedūrinė) yra pakankamai sudėtinga ir paini, todėl vartotojui sunkiai suprantama

2. Žemas abstraktumo lygis

Taikomosiose programose naudojamos išraiškos dažniausiai nusako ne pačią taisyklę, bet kaip ji turi būti vykdoma.

Pavyzdžiui, aukščiau pateikta taisyklė prieš ją įdiegiant galėtų būti papildyta taip:

```
Nuskaityti pirkėjo įrašą
KIEK=0
DO WHILE NOT END_OF_FILE
    KIEK=KIEK +1
    Nuskaityti pirkėjo įrašą
END
IF KIEK < 1000 THEN
    . . . (viskas, kas buvo aukščiau)
```

Iš pavyzdžio akivaizdu, kad taisyklės semantika (prasmė) susipina ir “atsiskiedžia” konkretaus taisyklės realizavimo detalėmis.

Be to, nuoroda į struktūrinį objektą - pirkėją - pašalinama. Vietoje to atsiranda programos kintamasis KIEK.

1. Taisyklių specifikaciją sunku skaityti ir suprasti, nes:

a) taisyklės išreiškiamos nurodant veiksmus, pranešimus apie klaidas bei jų vykdymo sąlygas;

b) galimos įvairiausios tos pačios taisyklės procedūrinių išraiškų formos;

c) taisyklės išreiškiamos detaliame taikymo abstrakcijos lygyje;

d) taisyklės išskaidytos per visą specifikaciją ir nėra pateiktos vienoje vietoje

2. Taisyklės sudėtinga patikrinti ir patvirtinti (verifikuoti ir validuoti), nes jos yra išbarstytos ir išreikštos procedūrinėje specifikacijos dalyje. Sunku patikrinti jų tarpusavio neprieštaringumą, pilnumą ir pateikti vartotojui, kad jis patvirtintų.

3. Kiekvienoje taikomojoje programoje yra taisyklių, kurios gali sukelti prieštaravimus tarp programų

- Procedūrinio taisyklių užrašymo būdo pagrindinė priežastis yra ta, kad jos apriboja galimas struktūrinių modelio sudėtinių dalių būsenas, tuo pačiu reglamentuodamos veiksmų ir pakeitimų rūšis, kurie leidžia keisti objektų būsenas.
- Sistemos realizavimo metu patogiau taisykles turėti “įmontuotas” į programą, o ne saugomas bendroje taisyklių bazėje. Todėl toks metodas ir tebenaudojamas specifikuojant taisykles

48. Taisyklių modeliavimo principai, struktūra ir klasifikavimas (10pask, 20 – 27psl.).

Šiuolaikiniai metodai leidžia bent dalinai išvengti aukščiau minėtų sunkumų, panaudodami šias priemones:

1. Deklaratyviai, o ne procedūriškai išreiškiama dauguma taisyklių.

2. Taisyklės išreiškiamos konceptualiaje abstrakcijos lygyje, panaudojant struktūrinės modelio komponentės sąvokas

3. Taisyklės atvaizduojamos atskirai nuo kitų specifikacijos dalių - centralizuotame taisyklių saugykloje

Nagrinėsime taisyklių aprašymo kalbos konstrukcijas, kurios sudarytos remiantis dažniausiai vartotojų reikalavimuose pasitaikančių taisyklių struktūra.

Taisyklių struktūrą nusako lentelėje pateikiamos klasifikacinės grupės, paskirstančios taisykles pagal apribojamus objektus bei pateikiamų apribojimų pobūdį

Objektas	Klasė			
	Kardi nalu mas	Aibė	Funk cija	Tvar ka
Atributas	1.1	1.2	1.3	1.4
Esybė	2.1	2.2	2.3	2.4
Esybių ryšiai	3.1	3.2.1, 3.2.2	3.3.1, 3.3.2	3.4
Esybės- atributo ryšiai	4.1	4.2.1, 4.2.2	4.3.1, 4.3.2	4.4

- Lentelės eilutės apibrėžia skirtingus DS struktūrinio modelio objektus, kurie gali būti taisyklių apribojami
- Stulpeliai nusako taisyklių klases, kurios įtakoja atitinkamus objektus
- Pavyzdžiui, pirmos eilutės taisyklės apriboja atributus, antros - esybes ir t.t.
- Yra keturios taisyklių klasės. Trys iš jų apriboja egzempliorių aibes, o kardinalumo tipo taisyklės nusako aibių savybes. Aibių elementai yra atributų arba esybių egzemplioriai, o aibės nusakomos atributo arba esybės vardu (arba šių vardų kombinacija)
- Galimas atvejis, kad ta pati taisyklė gali būti suformuluota esybei (2-a eilutė) arba šios esybės identifikatoriui, priklausomai nuo to, kaip ji išreikšta vartotojo reikalavimuose. Tokiu atveju priimta suformuluoti taisyklės esybei.

Pavyzdžiai

Atributo kardinalumo taisyklė (1.1 variantas)

- Ji apriboja tam tikro atributo reikšmių skaičių. Pavyzdžiui, galimų medžiagos kodų skaičius gali būti apribotas 1000. Tipinė šio apribojimo išraiška:
KARDINALUMAS (medžiagos-kodas) ≤ 1000

Atributo reikšmių aibės taisyklė (1.2 variantas)

- Ji apriboja atributo įgyjamas reikšmes. Pavyzdžiui, jei kainos atributas apribotas tarp 50 ir 200, tai galima užrašyti:
DIAPAZONAS (kaina) = [50 . . . 200]

Atributo reikšmės suformavimo taisyklė (1.3 variantas)

- Ji nusako funkciją, kuri leidžia nustatyti atributo reikšmę iš kitų reikšmių. Pavyzdžiui, jei atskaitymai į socialinio draudimo fondą gali būti nustatyti paėmus 3% gaunamo atlyginimo atributo reikšmės, tuomet tai galima užrašyti:
atskaitymai_soc_draudimui = atlyginimas * 0.03

Dalyvavimo ryšyje taisyklė (3.2.1 variantas)

- Ji apriboja ryšyje dalyvaujančių egzempliorių aibę. Ryšyje gali dalyvauti arba visi egzemplioriai, arba tik jų dalis. Pavyzdžiui, jei ryšyje “Žaliava -Tiekėjas” gali dalyvauti visi žaliavos egzemplioriai, tai galima išreikšti:

Žaliava : Tiekėjas = Žaliava

O jei ryšyje “Interesantai - Konkursas” gali dalyvauti tik asmenys iki 35 m., tuomet tai gali būti išreikšta:

Interesantai : Konkursas = Interesantai (amžius ≤ 35)

49. Bendri analizės principai (11pask, 2-4 psl.).

Bendri DS analizės principai, leidžiantys sudaryti korektišką ir pilną vartotojo poreikių specifikaciją:

1. Modeliuoti DS pilnai - modeliuoti detaliam konceptualiam abstrakcijos lygyje .
2. Viską išreikšti tiksliai - išvengti dviprasmiškumų, pertekliško ir prieštaravimo. Tai dažnai pasitaiko, ypač esant daug vartotojų.
Tam reikalinga speciali specifikavimo kalba. Tam tikrą specifikacijos korektiškumą, pavyzdžiui, esybių vardų unikalumo kontrolę gali užtikrinti specifikavimo priemonė, tačiau čia didelę reikšmę turi projektuotojo patirtis ir intuicija.
3. Viską išreikšti abstrahuotai - nepateikti specifikacijoje IS projektavimo bei programinės realizacijos detalių . Jos reikalingos vėlesniuose IS kūrimo etapuose. Tokiu atveju, specifikacija bus suprantama vartotojui.

Šį reikalavimą gali užtikrinti tik pats analitikas

4. Naudoti deklaratyvias išraiškas priemones. Vengti procedūrinių išraiškos formų.

Šį reikalavimą galima patenkinti naudojant deklaratyvaus tipo specifikavimo kalbą, leidžiančią neprocedūriškai atvaizduoti esybes bei taisykles.

5. Išvengti pertekliško - poreikių specifikacijoje konkrečius faktus pateikti vieną kartą.

Tai užtikrina tobulo specifikavimo priemonės

6. Viską išreikšti natūraliai - pilnai atspindėti realius DS objektus. Specifikavimo kalba turi užtikrinti adekvatų DS objektų atvaizdavimą. Specifikavimo kalbos pagrindu kuriamos konstrukcijos turi atitikti DS konstrukcijas

50. Analizės etapo sunkumai (11pask, 5-6 psl.).

Išskiriami du pagrindiniai sunkumai:

1. Specifikacijos kokybės užtikrinimas, išvengiant galimų klaidų, analizuojant informacinius poreikius.

Tai pasiekama naudojant abstraktesnius ir deklaratyvaus tipo modelius, kurie leidžia pilniau atspindėti specifines DS savybes. Šį uždavinį dalinai išsprendžia ER modelio bei deklaratyvaus taisyklių modelio panaudojimas.

Be to svarbu, kad analitiko klaidas arba neteisingai suprastus vartotojo poreikius būtų galima ištaisyti, panaudojant specifikacijos patvirtinimo metodus, pasitelkiant vartotoją.

2. Analizės etapo našumo užtikrinimas, ypač sunkiai pasiekiamas dėl poreikių pasikeitimų.

Vartotojo poreikių patikslinimas galėtų būti atliekamas būtent specifikacijos patvirtinimo etape, kuriam atlikti panaudotini efektyvūs validavimo metodai

51. Sistemų loginio ir fizinio projektavimo etapų sudėtis (11psl. 7-8 psl.)

