2.6 Metodologije in strategije konceptualnega načrtovanja

- Gradnja konceptualnega modela poteka postopoma, napredek izgradnje modela je odvisen od napredka v naši percepciji poslovne domene
- Zaznavanje najprej omogoči dojemanje v grobih obrisih, postopoma nato prehajamo v podrobnosti
- Posamezni načrtovalski koraki se izvajajo s pomočjo elementarnih transformacij sheme
- Transformacija sheme temelji na novem znanju o poslovni domeni, kar je rezultat procesa percepcije
- Poznamo različne strategije postopne gradnje konceptualnega modela: z vrha navzdol, od spodaj navzgor, od znotraj navzven, **po delih**

2.6 Metodologije in koncepti konceptualnega načrtovanja..

- Izhodiščna shema je entitetni tip, ki predstavlja shemo celotne poslovne domene (ali v nekem koraku sklop poslovne domene)
- Začetni vseobsegujoči entitetni tip po korakih razdelimo na vedno bolj elementarne, med seboj povezane koncepte
- Gre za postopno vse bolj natančno opisovanje poslovne domene do detajlov, ki zadoščajo za rešitev postavljenega problema
- Ustavimo se, ko je z detajli dosežen namen konceptualnega podatkovnega modela

Strategija Od spodaj navzgor

2.6 Metodologije in koncepti konceptualnega načrtovanja..

- Izhodiščno shemo tvori množica atributov, ki jih nato smiselno grupiramo, kreiramo entitene tipe (ki jim grupirani atributi pripadajo), entitetne tipe pa povezujem z razmerji in jih uvrščamo v hierarhijo tipov
- Atributi se grupirajo v skupine po svojem pomenu, vsaki skupini se priredi entitetni tip
- Ustavimo se, ko so ent. tipi primerno povezani med seboj za reševanje danega problema

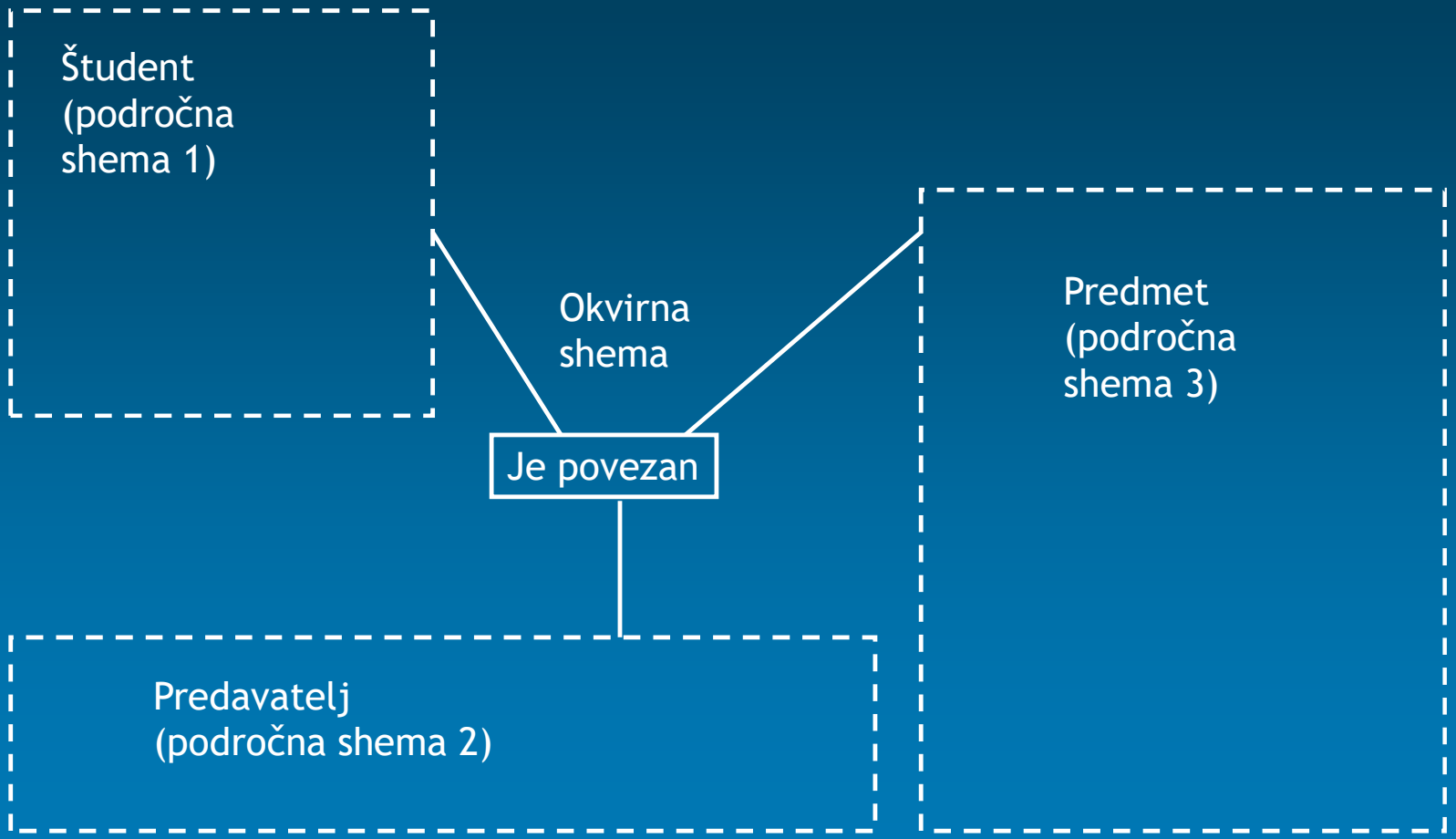
2.6 Metodologije in koncepti konceptualnega načrtovanja..

- Izhodiščno shemo tvori entiteti tip, ki ima glavno vlogo v poslovni domeni, ki mu nato po korakih postopoma dodajamo nove entitetne tipe in razmerja med njimi
- Ustavimo se, ko so ent. tipi primerno povezani med seboj za reševanje danega problema in smo dosegli dovolj detajlno predstavitev

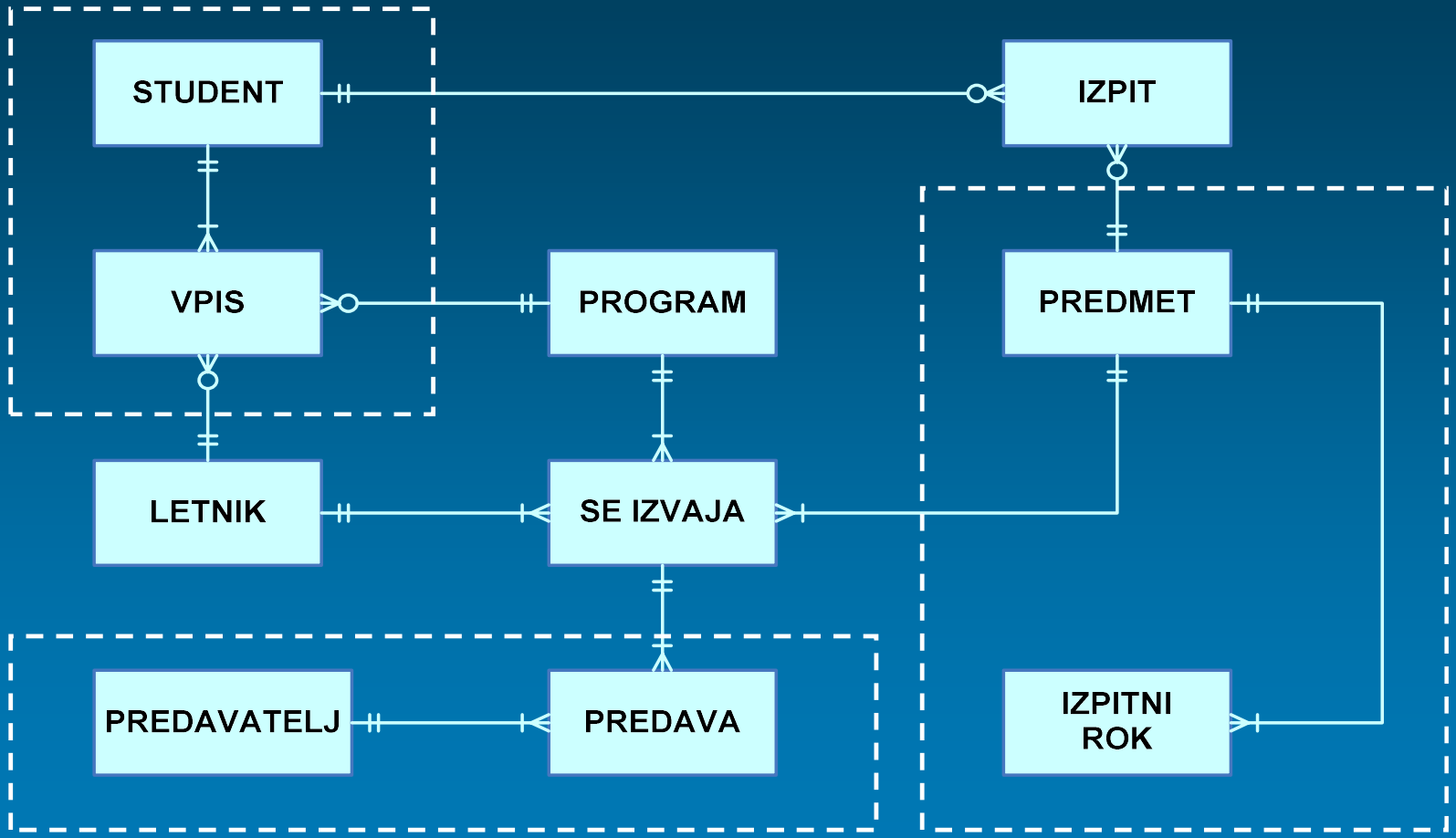
2.6 Metodologije in koncepti konceptualnega načrtovanja..

- V praksi najbolj uporabna strategija
- Najprej kreiramo okvirno shemo z najpomembnejšimi entitetnimi tipi in razmerji med njimi
- Nato shemo razdelimo v posamezne dele (sklope, področja, podmodele), kjer so jedra prav ti tipi
- Tako modeliranje razdelimo v več lažje obvladljivih delov, kjer so vključeni poznavalci posameznih področij
- Poleg tega lahko modeliranje izvajamo tudi vzporedno
- Posamezne sklope na koncu združimo v eno shemo s pomočjo okvirne sheme

2.6 Načrtovanje po delih



2.6 Načrtovanje po delih



2.6 Metodologije in koncepti konceptualnega načrtovanja..

- V nadaljevanju bo podanih nekaj priporočil za področje izdelave konceptualnega podatkovnega modela
- Z njihovo dosledno uporabo si zagotovimo maksimalno stopnjo normalnosti (vsaj 3. NO) dobljenih relacij (tabel).

2.6 Metodologije in koncepti konceptualnega načrtovanja..

- Načeloma naj entitentni tipi vsebujejo samo lastne atribute!
- Lastnost opazovanega entitetnega tipa lahko opišemo:
  - Z določitvijo atributa
  - Z uvedbo novega entitetnega tipa in vzpostavitvijo razmerja med njima
- Primer: Kraj stalnega prebivališča za OSEBA
  - Določimo atribut *Stalno prebivališče* **ali**
  - Uvedemo entitetni tip KRAJ in razmerje *Stalno prebivališče* med entitetnima tipoma OSEBA in KRAJ

2.6 Metodologije in koncepti konceptualnega načrtovanja..

- Določitev atributa pomeni, da "želimo", da bo za opazovano lastnost za poljubno entiteto opazovanega entitetnega tipa možna katerakoli tipu atributa (in dolžini) ustrezna vrednost
- Uvedba novega entitetnega tipa in razmerja pomeni, da "želimo", da bo za opazovano lastnost za poljubno entiteto opazovanega entitetnega tipa možna le vrednost, ki se nahaja v entitetni množici na novo uvedenega entitetnega tipa

2.6 Metodologije in koncepti konceptualnega načrtovanja..

- Razmerje tipa M:N odpravimo preko uvedbe "vmesnega" entitetnega tipa in ustreznih razmerij
- Razmerij tipa M:N ni potrebno odpraviti vedno, potrebno jih je odpraviti takrat, ko ima razmerje tudi svoje atribute
- Vmesni entitetni tip ima vedno določen pomen, zato je zaradi večje berljivosti, izraznosti in informativnosti konceptualnega modela priporočljivo odpraviti vsa razmerja tipa M:N v modelu
- Razmerje tipa M:N se sicer odpravi avtomatsko pri avtomatskem prehodu na logični model oz. pri kreiranju logičnega podatkovnega modela na podlagi konceptualnega

2.6 Metodologije in koncepti konceptualnega načrtovanja..



2.6 Metodologije in koncepti konceptualnega načrtovanja..

- Standarde si določi/oblikuje razvojna ekipa
- Primer:
  - Entitetnim tipom in atributom dajemo opisna imena
  - Imena entitetnih tipov naj bodo vedno bodisi v ednini, bodisi v množini
  - Enotni standardi poimenovanja atributov:
    - Identifikator ID_
    - Zapordna številka ZAP_ST_
    - Ime (entitete) IME_OSEBE, IME_PODJETJA, IME_PSA
    - …
  - Atributa lahko imata ISTO ime samo v primeru, ko se njun pomen v popolnosti ujema!
  - **Istoimenski** atributi morajo biti tudi **istopomenski**!

2.7 Načrtovanje zunanjih shem..

- Zunanja shema ali uporabniški pogled:
  - je pogled na konceptualni podatkovni model oz. izsek konceptualnega podatkovnega modela. Izsek je lahko narejen na podlagi funkcionalnega področja, lahko pa predstavlja le pogled na celoten konceptualni podatkovni model s strani posameznega uporabnika
- Zunanja shema ali uporabniški pogled dejansko predstavlja konceptualni podatkovni podmodel
- Konceptualni podatkovni podmodel nastane na podlagi zajetih uporabniških zahtev

2.7 Načrtovanje zunanjih shem..

- Uporabniške zahteve predstavljajo vir (podlago) za pridobitev podatkov o poslovnih procesih, poslovnih funkcijah in poslovnih pravilih. Splošneje lahko govorimo o virih.
- Pri večjih projektih načrtujemo po delih - zunanjih shemah posameznih uporabnikov
- Oblike uporabniških zahtev - virov so:
  - Formalni in neformalni pogovori
  - Dokumentacija
  - Opis v naravnem jeziku
  - Obrazci
  - "Stare aplikacije"

2.7 Načrtovanje zunanjih shem..

- Predstavniki naročnika/uporabnika praviloma ne poznajo principov konceptualnega podatkovnega modeliranja
- V pogovorih s predstavniki naročnika/uporabnika moramo vprašanja zastavljati tako, da bodo odgovori omogočili izdelavo semantično ustreznega konceptualnega podatkovnega (pod)modela
- Vprašanja oz. odgovori morajo omogočiti, da iz njih izluščimo entitetne tipe, razmerja, atribute, kardinalnosti in mandatornosti (za atribute in razmerja)
- Pogovori (lahko) potekajo v več iteracijah, kjer iz iteracije v iteracijo vse bolj posegamo v podrobnosti

2.7 Načrtovanje zunanjih shem..

- V dokumentacijo sodijo različne oblike zapisov o pravilih, dejstvih, dogodkih, osebah, objektih, procesih in ostalem o poslovni domeni oz. področju poslovne domene
- Vrste – tipi dokumentacije (za poslovni problem, podatkovno bazo, obstoječe rešitve):
  - Poslanstvo podjetja, strateški načrt, organizacijska shema
  - Priročniki, navodila, zapisniki sestankov, pritožbeni zapisniki
  - Razpisna dokumentacija, zakonodaja
  - Interni pravilniki in predpisi
- Dokumentacija se lahko nahaja v elektronski ali papirni obliki

2.7 Načrtovanje zunanjih shem..

- Opis v naravnem jeziku uporabniki praviloma naredijo na zahtevo razvijalcev, ki iz dokumentacije, obrazcev in starih aplikacij ne morejo dobiti vse potrebne informacije
- Pridobimo na osebnem nivoju od dejanskih uporabnikov: pogovori ali izpolnjevanje vprašalnikov
  - Pogovori: prost opis problema, vključitev uporabnika v načrtovanje, sprotno komentiranje in preverjanje dejstev, prilagajanje pogovora uporabniku. Vendar: zamudno, uspeh odvisen od spretnosti in pripravljenosti sogovornikov
  - Vprašalniki: hiter zajem odgovorov večjega števila uporabnikov, možni fiksni in odprti odgovori, anonimnost, enostavno tabeliranje in analiza odgovorov. Vendar: možnost slabega odziva, delno izpolnjenih vprašalnikov, fiksna oblika vprašalnika.

2.7 Načrtovanje zunanjih shem..

- Načrtovalec pridobi na ne-osebnem nivoju
- Raziskava problema
  - Literatura: knjige, revije, Internet, obstoj delnih rešitev problema
- Opazovanje delovanja organizacije
  - validacija doslej zbranega znanja
  - ustvarjanje slike celote iz kompleksne množice delčkov
- Predmet opisa v naravnem jeziku je lahko na različnem nivoju detajlov:
  - Celotna poslovna domena, funkcionalno področje ali sklop
  - Poslovni proces (poslovne funkcija)
  - Poslovno pravilo (postavljanje različnih omejitev)

2.7 Načrtovanje zunanjih shem..

- Obrazci predstavljajo podlago za vnos podatkov in niso sami sebi namen. V večini primerov bo vnos podatkov iz izpolnjenih obrazcev v podatkovno bazo prevzela določena poslovna funkcija
- Šifriranje obrazcev in podatkovni slovar
- Obrazci imajo praviloma več logičnih področij-sklopov
- Javna uprava: odprava ročnega izpolnjevanja obrazcev

2.7 Načrtovanje zunanjih shem..

- Za načrtovanje na osnovi obrazcev lahko "lokalno" uporabimo strategijo po delih:
  - Določimo logična področja na obrazcu
  - Za vsako področje identificiramo entitetne tipe, razmerja in atribute
  - Zgradimo okvirno shemo
  - Zgradimo področne sheme
  - S pomočjo okvirne sheme povežemo področne sheme med seboj

2.7 Načrtovanje zunanjih shem..

- Obravnavana oblika je na voljo le v primeru prenove informacijskega sistema oz. podatkovne baze
- Možne oblike dokumentacije starih aplikacij:
  - Podatkovni model: konceptualni, logični (relacijski), fizični
  - Funkcionalni model: dekompozicijski diagram in opisi funkcij
  - Model poslovnih pravil: opisi poslovnih pravil (omejitev)
  - Uporabniška dokumentacija
  - Aplikacija sama
- Na posamezne vnosne in pregledovalne forme stare aplikacije lahko v mejnem primeru gledamo kot na obrazce in jih pri izdelavi konceptualnega podatkovnega modela nove aplikacije kot take tudi uporabimo

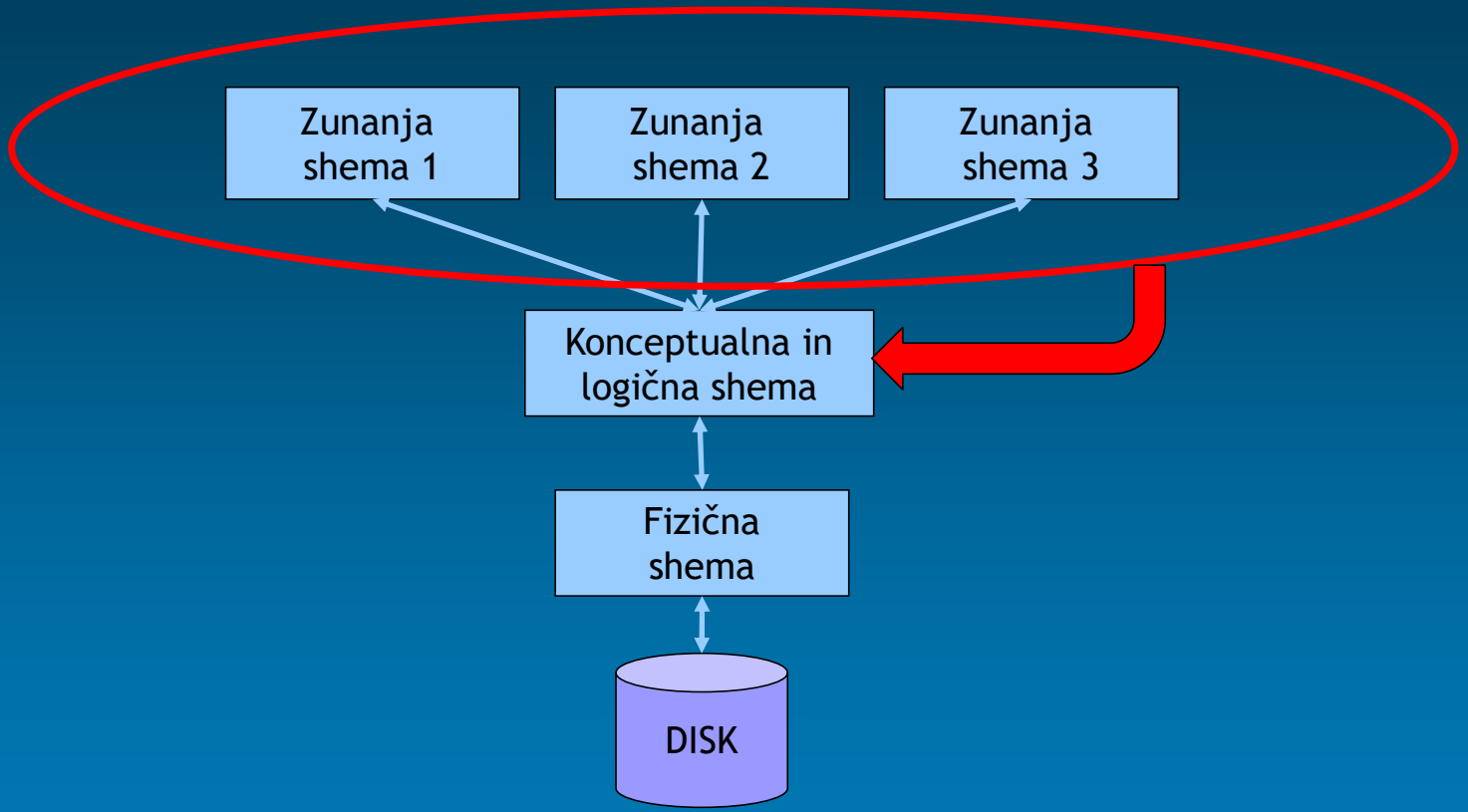
2.7 Načrtovanje zunanjih shem..

- Tipična uporaba strategije *Po delih*
- Določitev področij praviloma po funkcionalnih področjih, lahko pa tudi drugače
- Kombinirana uporaba virov (uporabniških zahtev) pri izdelavi podmodelov
- Praviloma na podlagi pregledane dokumentacije, obrazcev in stare aplikacije zahtevamo potrjene opise v naravnem jeziku in sestanke (formalni in neformalni pogovori)
- Iterativen postopek

2.8 Združevanje zunanjih shem..

- Konceptualnega modela ni možno pridobiti kot enostavno združitev konceptualnih podatkovnih podmodelov
- Med posameznimi podmodeli obstaja določen presek, ki omogoča združevanje oziroma povezovanje shem (podmodelov) v celovit konceptualni model

2.8 Združevanje zunanjih shem..



2.8 Združevanje zunanjih shem..

- Pred združevanjem podmodelov je potrebno odpraviti morebitna neskladja med njimi
- Neskladja so posledica razlik v percepciji s strani različnih uporabnikov in analitikov: pravilnosti, različnosti in podrobnosti zaznavanja (razumevanja)
- Vrste neskladij:
  - Imenska neskladja
  - Različne perspektive
  - Semantično ekvivalentni deli podmodela
  - Nezdružljive integritetne omejitve

  } Pomenska neskladja

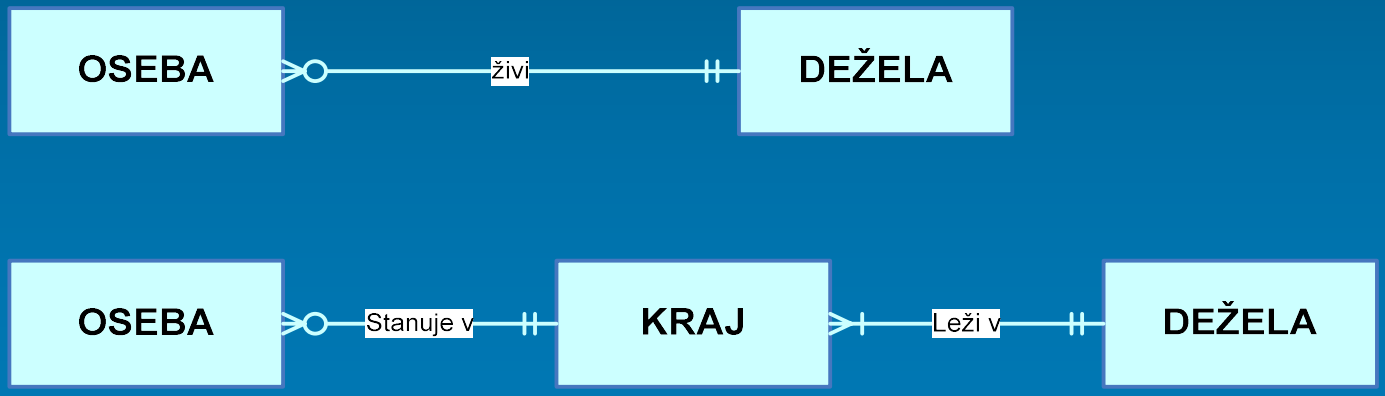
- Analiza neskladij je prvi korak združevanja podmodelov

2.8 Združevanje zunanjih shem..

- Neskladja na področju treh vrst konceptov: entitetnih tipov, atributov in razmerij
- Sinonimi: isti koncepti so različno poimenovani
  - Primer za entitetni tip - sinonimi: PROFESOR, PREDAVATELJ
  - Primer za atribut - sinonimi: Naslov, Ulica
  - Primer za razmerje – sinonimi: Stanuje, Stalno prebivališče
- Homonimi: enako poimenovani koncepti, ki imajo dejansko različen pomen (različnost zaznavanja in/ali pravilnost zaznavanja)
  - Primer za razmerje – homonimi: Stanuje; stalno, začasno
- Za ugotavljanje neskladij je zelo pomembno, da so entitetni tipi, razmerja in atributi podrobno opisani

2.8 Združevanje zunanjih shem..

- Posledica različne podrobnosti zaznavanja in različnosti samega zaznavanja po področjih
- Različna percepcija poslovnih procesov, poslovnih funkcij in poslovnih pravil, predvsem nivo podrobnosti po področjih



2.8 Združevanje zunanjih shem..

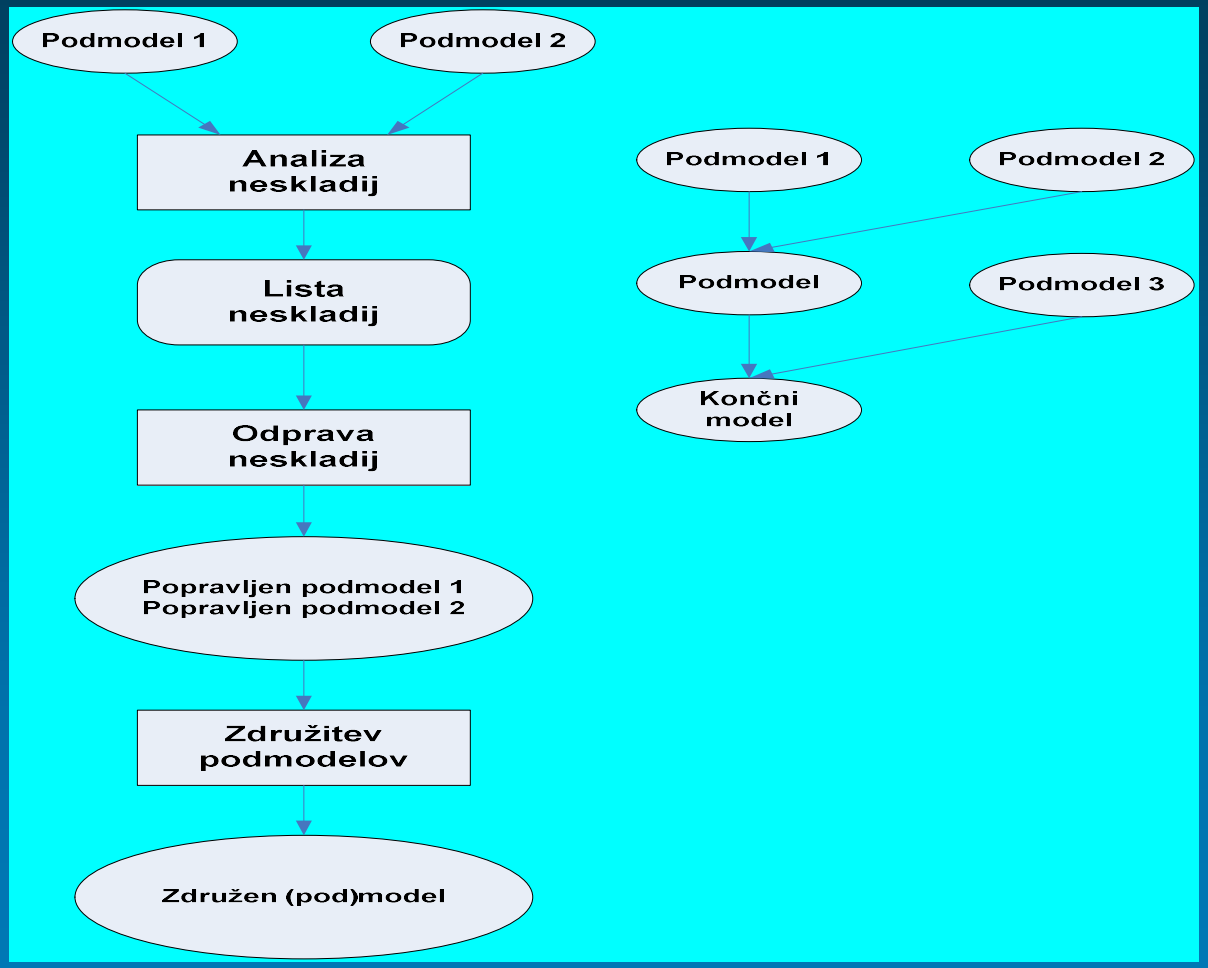
- Posledica dejstva, da posamezne dele modela lahko predstavimo na semantično ekvivalentne načine
- Pogost primer je predstavitev iste lastnosti na eni strani z atributom, na drugi strani pa z entitetnim tipom in razmerjem

2.8 Združevanje zunanjih shem..

- Različne kardinalnosti (istopomenskih) razmerij med (istopomenskima) entitetnima tipoma
- Različne mandatornosti (istopomenskih) atributov v okviru entitetnega tipa
- Različne mandatornosti (istopomenskih) razmerij med (istopomenskima) entitetnima tipoma

Združevanje zunanjih shem oz. podmodelov

2.8 Združevanje zunanjih shem..



2.8 Združevanje zunanjih shem..

- Združevanje podmodelov je iterativen postopek, kjer v posameznem koraku združimo po dva podmodela
- Trije načini združevanja:
  - Preko skupnega entitetnega tipa
  - S pomočjo hierarhije tipov
  - S pomočjo razmerja med entitetnima tipoma (podmodeloma)

- V različnih podmodelih se lahko gradniki pojavijo tudi kot
  - homonimi (enako ime, različen pomen); npr. ime (kraja) in ime (osebe)
  - sinonimi (različno ime, enak pomen); npr. učitelj in profesor

2.8 Združevanje zunanjih shem..

- Združimo skupna entitetna tipa.
  - Entitetna tipa imata lahko enako ime. Ugotoviti je potrebno, če ne gre za homonim (enako ime, različen pomen). Če gre za homonim, potem je enega od entitetnih tipov potrebno preimenovati. Takšen entitetni tip ne more združiti dveh podmodelov.
  - V primeru, da ni entitetnih tipov z enakima imenoma, iščemo sinonime (različno ime, enak pomen).
  - Če za dva entitetna tipa ugotovimo, da sta pomensko enaka oz. da gre za sinonim, potem je enega od entitetnih tipov potrebno preimenovati. Ta korak omogoči združitev podmodelov takrat, ko na prvi pogled združitev ne bo možna. Shemi lahko združimo preko tega entitetnega tipa

2.8 Združevanje zunanjih shem..

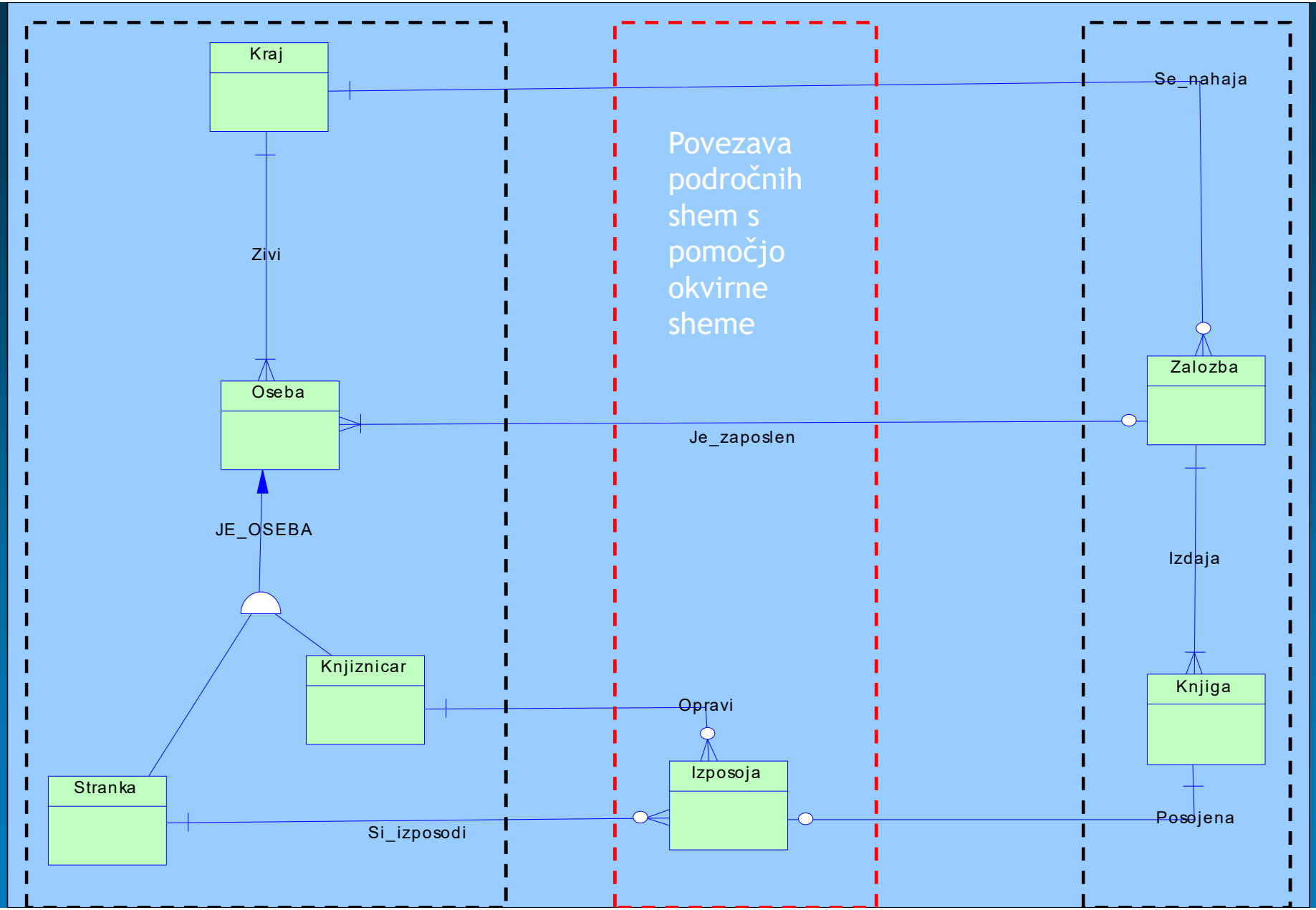
- Za entitetni tip kot vezni člen moramo združiti še atribute, kar izvedemo preko operacije unije
- Unija ne more biti povsem avtomatična. Tudi za združitev atributov je potrebno ugotoviti homonime in sinonime ter ustrezno ukrepati
- Entitetni tip vezni člen “prevzame” vsa razmerja, ki so bila pred združitvijo vezana na oba entitetna tipa
- Določiti je potrebno še (novi) enolični identifikator

2.8 Združevanje zunanjih shem..

- V okviru združevanja preko skupnega entitenega tipa lahko ugotovimo, da dva različno poimenovana entitetna tipa nista sinonima, temveč obstaja med njima (doslej še neodkrita) hierarhija tipov
- Podmodela lahko tako povežemo preko vpeljave hierarhije tipov
- Po vpeljavi hierarhije tipov je potrebno ugotoviti homonime in sinonime med atributi ter izvesti morebitna potrebna preimenovanja
- V podtipu odstranimo enako imenovane (in isto-pomenske) atribute, ki jih le-ta podeduje od nadtipa

2.8 Združevanje zunanjih shem..

- V okviru združevanja preko skupnega entitenega tipa lahko ugotovimo, da ni niti skupnih entitetnih tipov, niti ni možno vpeljati hierarhije tipov. Obstaja pa (morda doslej še neodkrito) razmerje med entitenim tipom prvega podmodela in entitetnim tipom drugega podmodela
- Podmodela lahko združimo preko na novo določenega razmerja


Kraj
Se_nahaja
Povezava
področnih
shem s
pomočjo
okvirne
sheme
Zivi
Oseba
Zalozba
Je_zaposlen
JE_OSEBA
Izdaja
Knjiznicar
Knjiga
Opravi
Izposoja
Stranka
Posojena
Si_izposodi

2.8 Združevanje zunanjih shem..

- Kdaj (v katerem trenutku) združevati podmodele? Dve skrajnosti:
  - Prva možnost: podmodele izdelamo do konca in potem izvajamo predstavljene postopke združevanja
  - Druga možnost: preden v kateregakoli od podmodelov dodamo nov entitetni tip, smiselno izvedemo (uporabimo) predstavljene načine združevanja shem in odpravljanje neskladij
- V praksi delujemo nekje vmes med obema možnostima

2.8 Združevanje zunanjih shem..

- Zelo pomembno je, da se pri izdelavi modela uporablja orodje CASE (npr. Power Designer), ki ima na voljo repozitorij, v katerem so opisani vsi gradniki vseh podmodelov

- Ažurna raba repozitorija omogoča sprotno souporabo konceptov in odkrivanje neskladij

2.9 Celovit pristop h konceptualnemu načrtovanju z orodji CASE (npr. PowerDesigner)

- Operativni koraki konceptualnega načrtovanja:
  - K1.1: Identificiraj entitetne tipe
  - K1.2: Identificiraj povezave
  - K1.3: Identificiraj in z entitetnimi tipi poveži atribute
  - K1.4: Atributom določi domene
  - K1.5: Določi kandidate za identifikatorje, med njimi izberi najprimernejšega (bodoči primarni ključ)
  - K1.6: Po potrebi uporabi elemente razširjenega diagrama entiteta – razmerje (dodatne hierarhije)
  - K1.7*: Preveri, če v modelu obstajajo odvečni elementi
  - K1.8*: Preveri, če model "zdrži" transakcije
  - K1.9*: Preveri model z uporabnikom

} Gradnja konceptualne sheme na podlagi uporabniških zahtev z uporabo primerne strategije

} Dodelav detajlov konceptualne sheme

} Validacija konceptualne sheme

Korake, označene z zvezdicami si bomo podrobneje pogledali.

Primer: spletna knjigarna (K1.1-K1.6)

- **Uporabniške zahteve:**
  - Registriracija strank z imenom, naslovom, e-pošto in telefonsko številko. Omogočen je pregled knjig, oddaja naročil in sledenje zgodovini naročil.
  - Knjigarna mora vzdrževati podatke o knjigah, vključno z naslovom, avtorji, ISBN, založnikom, datumom objave, žanrom, ceno in količino na zalogi.
  - Stranke imajo možnost oddaje naročil, ki vsebujejo več knjig. Vsako naročilo mora imeti datum naročila, naslov za dostavo in status naročila (npr. v čakanju, odposlano, dostavljeno).
  - Knjigarna hrani podrobne podatke o avtorjih, kot so njihovo ime, biografija in datum rojstva.
  - Knjigarna hrani tude podatke o založnikih, vključno z njihovim imenom in kontaktnimi podatki.

Spletna knjigarna – entitetni tipi in atributi

Stranka: IDstranke (PK), Ime, Naslov, Email, Telefon

Knjiga: IDknjige (PK), Naslov, ISBN, DatumObjave, Cena, KoličinaNaZalogi, Žanr

Naročilo: IDnaročila (PK), DatumNaročila, NaslovZaDostavo, StatusNaročila

Avtor: IDavtorja (PK), Ime, Biografija, DatumRojstva

Založnik: IDzaložnika (PK), Ime, KontaktniPodatki

- Razmislite o atributih, njihovih domenah in kardinalnostih.
- Razmislite o primarnih ključih.

Spletna knjigarna – razmerja in hierarhije

- **Stranka** lahko odda več **Naročil** (ena-proti-mnogo).
- **Naročilo** lahko vsebuje več **Knjig** (mnogo-proti-mnogo).
- **Knjigo** lahko napiše več **Avtorjev** (mnogo-proti-mnogo).
- **Knjigo** izda en **Založnik** (ena-proti-mnogo).

- Razmislite o kardinalnosti razmerij.
- Kaj je **Naročilo**?
- Ali lahko vpeljemo kakšno hierarhijo?

2.10 Kvaliteta konceptualnega modela..

- Semantična pravilnost konceptualnega modela še ne zagotavlja njegove kvalitete
- Tudi semantično pravilni modeli so lahko boljši ali slabši, bolj ali manj razumljivi
- Kvaliteto konceptualnega podatkovnega modela ocenjujemo po naslednji karakteristikah:
  - Popolnost
  - Pravilnost
  - Minimalnost
  - Izraznost
  - Berljivost

2.10 Kvaliteta konceptualnega modela..

- Konceptualni model je popoln, če predstavlja vse pomembne lastnosti poslovne domene.
- Popolnost preverimo tudi tako, da za vsako uporabniško zahtevo preverimo, ali je na modelu predstavljena.
- Pravimo, da je model popoln z ozirom na uporabniške zahteve

2.10 Kvaliteta konceptualnega modela..

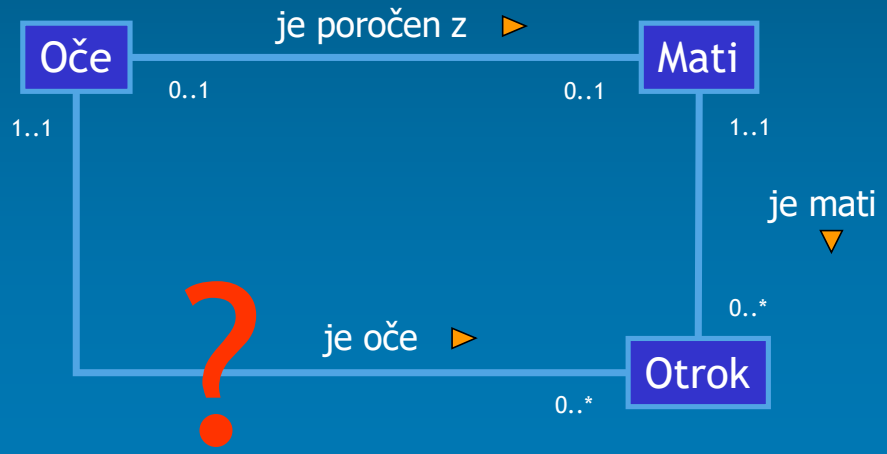
- Model je sintaktično pravilen, če je izdelan v skladu s sintaktičnimi pravili entitetnega modela (uporaba CASE orodja)
- Model je semantično pravilen, če vsi atributi in razmerja vseh entitetnih tipov ustrezajo poslovni domeni in njenim potrebam

2.10 Kvaliteta konceptualnega modela..

- Model je minimalen, če nima redundantnih predstavitev problemskega področja oz. če noben koncept ni redundantno predstavljen
- Model je torej minimalen, če iz njega ne moremo odstraniti nobenega koncepta, ne da bi pri tem ne da bi pri tem model izgubil popolnost

2.10 Kvaliteta konceptualnega modela

- K1.7 - Preverimo, če v modelu obstajajo odvečni elementi:
  - Pregledamo povezave 1 – 1 (po možnosti združimo povezana entitetna tipa)
  - Odstranimo odvečne povezave (npr. tranzitivne), s tem da preverimo tudi "časovni okvir" razmerja (in posledično njegovo semantiko)



Izraznost

2.10 Kvaliteta konceptualnega podatkovnega modela..

- Model je izrazen, če ga je možno doumeti in razumeti samo po sebi, ne da bi ga bilo potrebno dodatno opisovati, pojasnjevati in komentirati
- Dva modela sta lahko semantično enaka, vendar je eden lahko bolj izrazen od drugega

2.10 Kvaliteta konceptualnega podatkovnega modela

- Večjo berljivost sheme dosežemo preko naslednjih pravil:
  - Približno enaka velikost simbolov za entitetne tipe
  - Barvanje entitetnih tipov po določenem kriteriju
  - Poudariti simetričnost in hierarhičnost diagrama oz. izdelovati ga v tej smeri
  - Povezave naj se ne križajo, če je to le mogoče
  - Pri hierarhiji tipov naj bodo nadtipi zgoraj, podtipi pa spodaj
- Izraznost in berljivost ugodno vplivata na informativnost modela (povezava med podatki in informacijo)

K1.8 - Preverjanje transakcij

- V tem kontekstu izraz „transakcija“ pomeni poljubno opravilo, ki ga bomo izvajali nad PB

- Preveriti moramo če model podpira vse zahtevane transakcije.
  - Transakcije izvajamo “ročno”
  - Če neke transkacije ne uspemo izvesti, je model pomanjkljiv (manjka bodisi entitetni tip, povezava ali atribut)
- Možna dva pristopa:
  - Preverjanje opisa transakcij
  - Preverjanje transakcijskih poti

Preverjanje transakcij

- Preverjanje opisa transakcij
  - Vsako transakcijo opišemo;
  - Preverimo, če model zajema vse entitetne tipe, povezave in atribute, ki jih transakcija potrebuje.

Preverjanje transakcij

- Primer opisa transakcijskih zahtev
  - Vnos podatkov:
    - Vnesi podatke o študentih (npr. 24010637, Monika Jemec,...)
    - Vnesi podatke o predmetih (npr. 70029, Razvoj IS, Letni,...)
    - ...
  - Urejanje in brisanje podatkov:
    - Uredi/briši podatke o študentu
    - Uredi/briši podatke o predmetih
    - ...
  - Poizvedbe
    - Izpiši vse študente, ki so se vpisali v določen letnik, določene smeri, določenega programa
    - Izpiši vse predmete, ki jih je opravil določen študent
    - ...

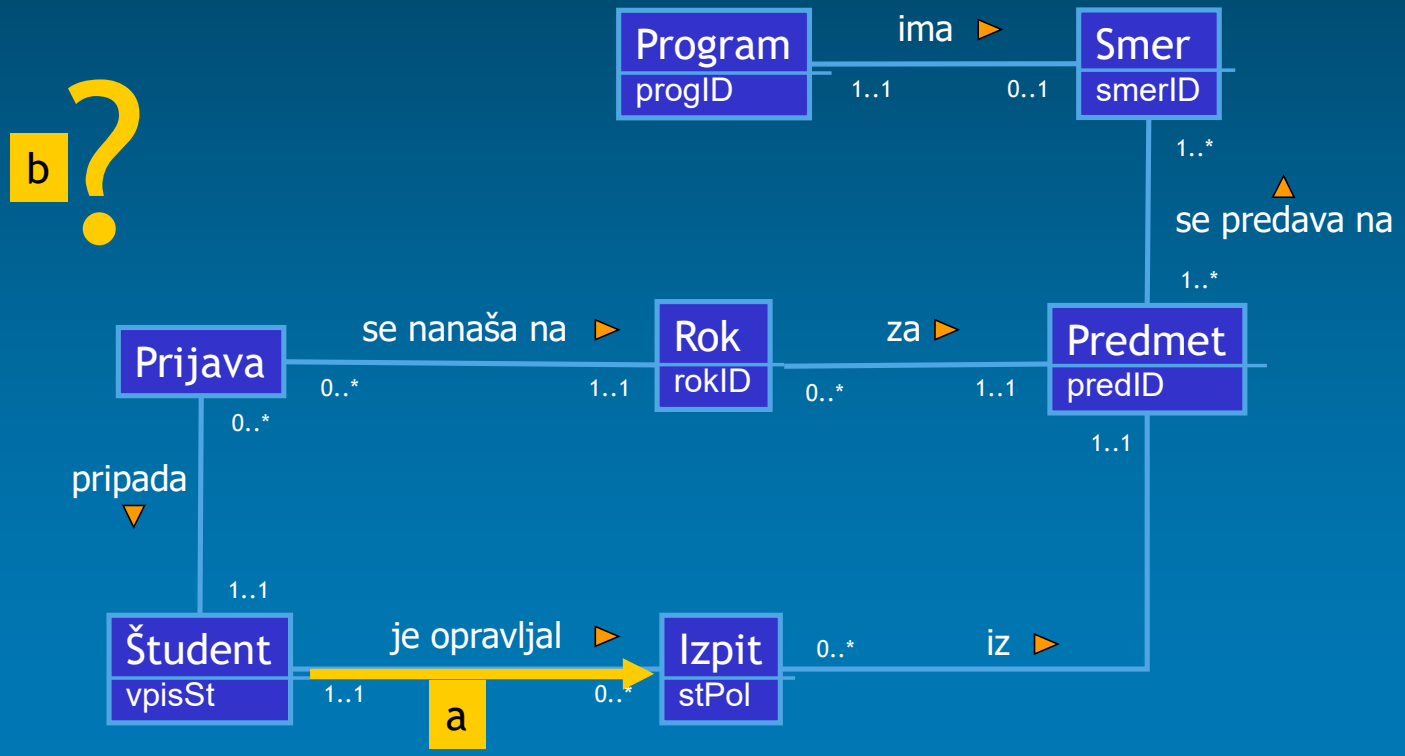
Preverjanje transakcij

- Preverjanje transakcijskih poti
  - Transakcije preverimo na modelu – pot transakcije narišemo
  - Pristop načrtovalcu omogoča:
    - Da identificira pomanjkljivosti modela (če pot za neko transkacijo ni možna)
    - Da identificira dele modela, ki so transakcijsko kritični
    - Da odkrije odvečne dele modela (deli, ki jih ne potrebuje nobena transakcija)

- **Preverjanje transkacij je zamudno vendar pomembno delo!!**

Primer preverjanja transakcij

a) Izpiši vse predmete, ki jih je opravil določen študent
b) Izpiši vse študente, ki so se vpisali v določen letnik, določene smeri, določenega programa



K1.9 - Preverjanje celotnega modela

- Na koncu model preverimo z uporabnikom
- Anomalije, pomanjkljivosti, napake,... lahko vodijo v ponovitev načrtovalskih korakov
- V praksi pogosto zahtevamo, da uporabnik oz. naročnik podpiše podatkovni model (user sign-off)