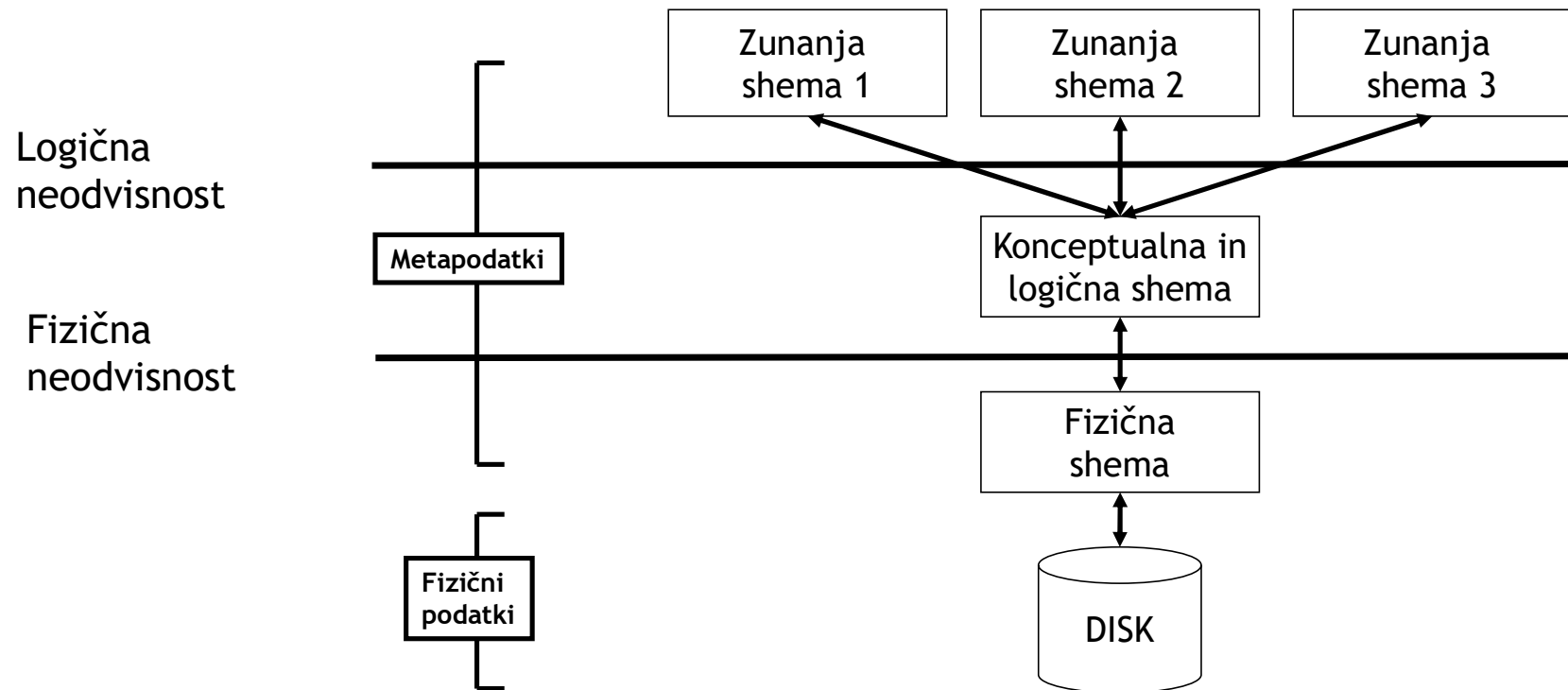


Poglavje 1

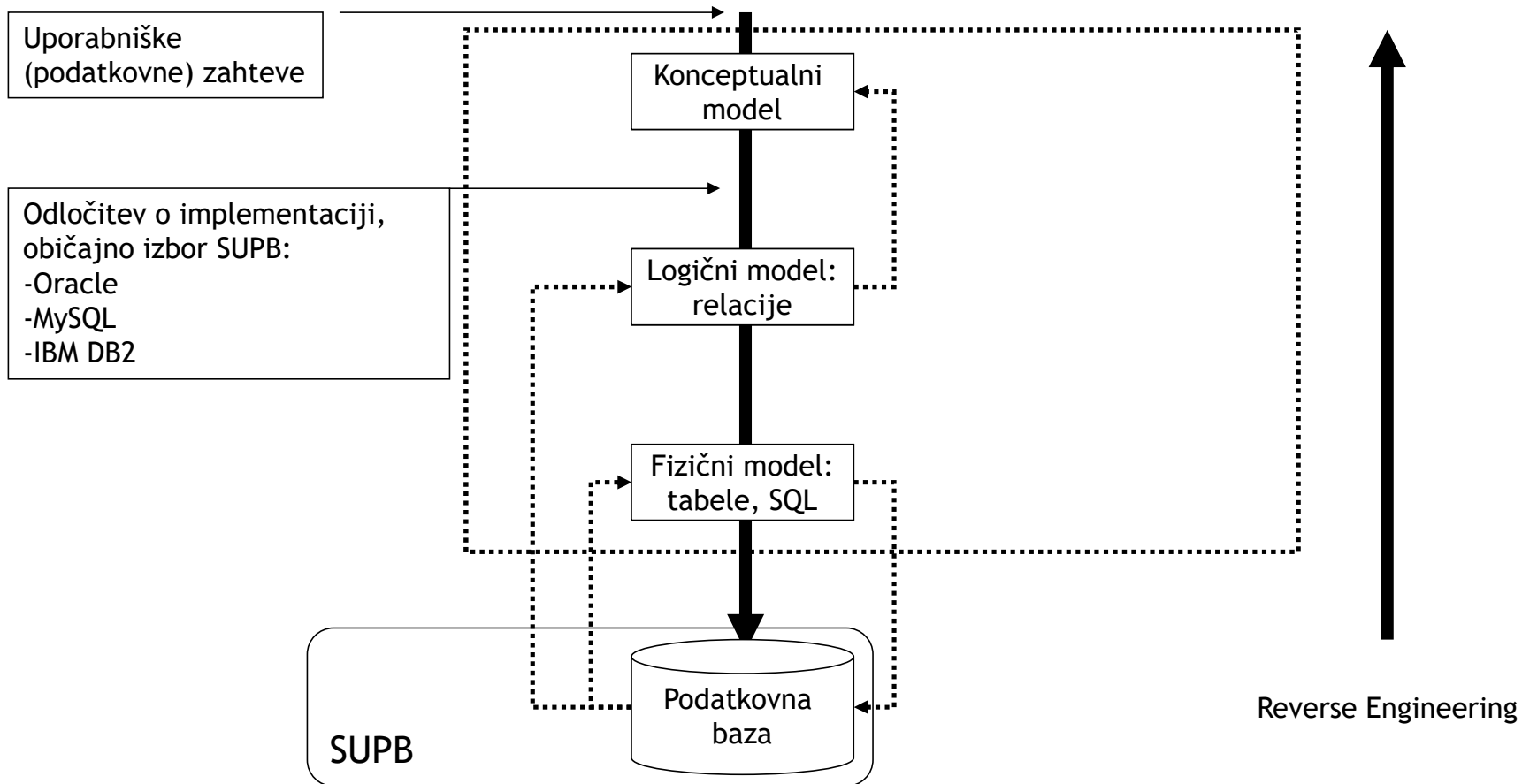
**Konceptualno načrtovanje
transakcijskih podatkovnih baz**

Trinivojska predstavitev podatkov...

- Podatki so v PB opisani na treh nivojih:
 - Zunanje sheme (zunanji, uporabniški nivo)
 - Konceptualna (logična) shema
 - Fizična (notranja) shema



1.1 Trije nivoji načrtovanja – trije modeli..



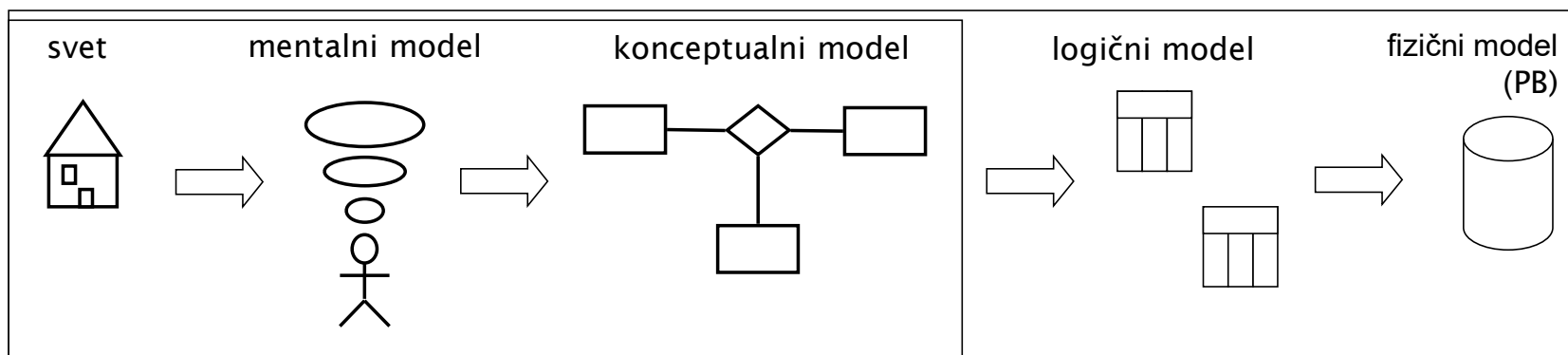


Konceptualni podatkovni model

- Formalizem s katerim opišemo, kaj bi želeli hraniti v PB ter kakšne povezave obstajajo med elementi, ki jih želimo hraniti, se imenuje konceptualni model.
- Konceptualni model je torej način, kako na visoki ravni abstrakcije razumljivo in ne preveč tehnično opišemo podatke, ki jih želimo hraniti ter skrijemo nepomembne podrobnosti.
- Konceptualni model odraža uporabnikovo zaznavanje realnega sveta oziroma poslovnega problema, ter načrtovalčevo videnje problema (stična točka).

1.1 Trije nivoji načrtovanja – trije modeli

- Načrtovanje PB: od realnega sveta do fizične podatkovne baze za potrebe poslovne domene



1.2 Kaj je konceptualno načrtovanje?..

- Konceptualno načrtovanje je postopek opredelitve podatkovnih potreb oz. zahtev (poslovne ali druge) domene s pomočjo konceptualnega modela
- Konceptualno načrtovanje preko konceptualnega modela poskrbi za opis pomena podatkov, potrebnih za poslovno domeno
- Konceptualno načrtovanje je neodvisno od dejanskega podatkovnega modela (relacijski, ...)
- Konceptualnega načrtovanja ne moremo avtomatizirati, za njegovo izvedbo je odgovoren analitik. Gre za prenos semantike v model.

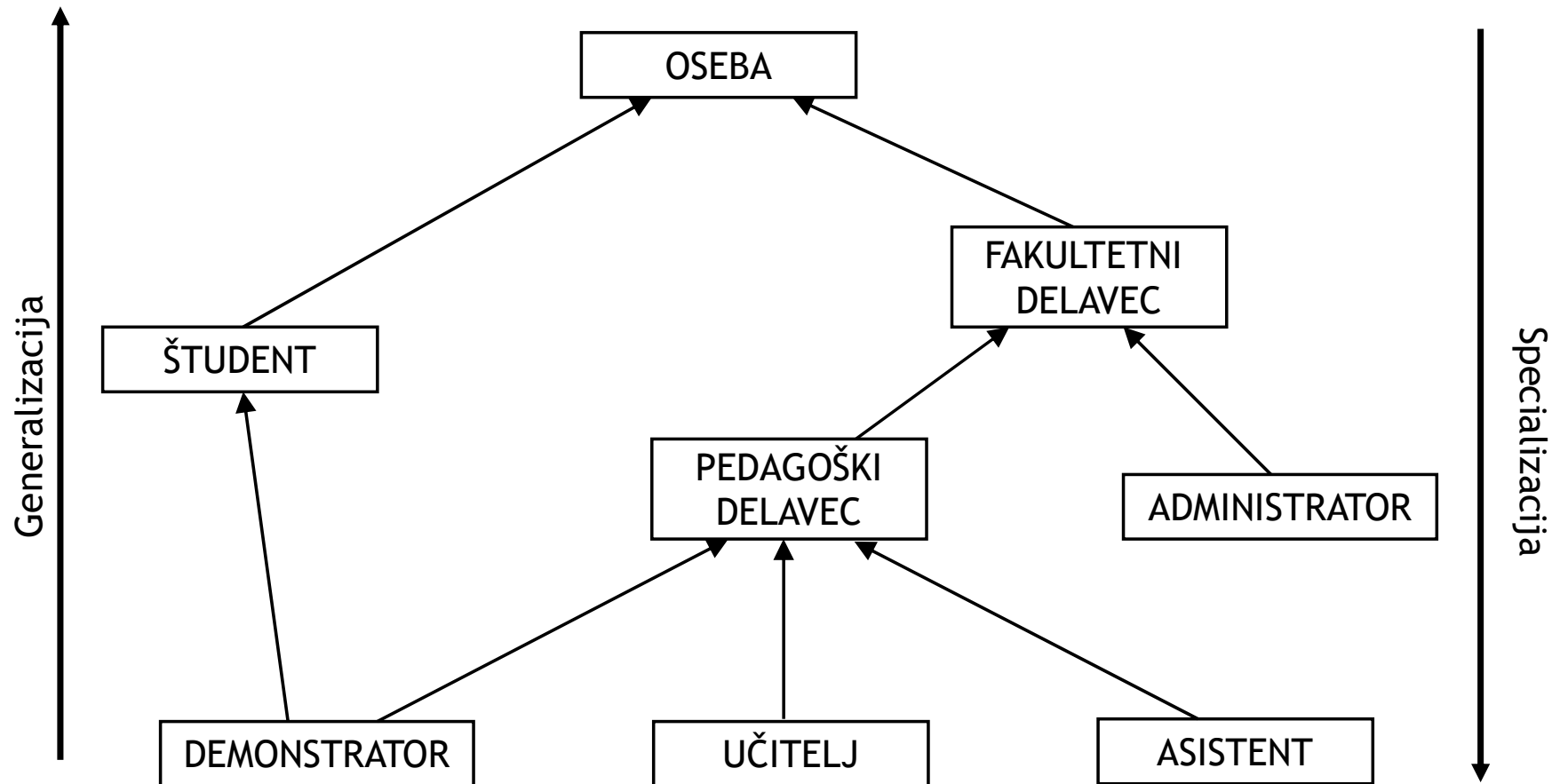
1.2 Kaj je konceptualno načrtovanje?

- Je najbolj kritično, saj se napake narejene pri konceptualnem načrtovanju prenašajo naprej na naslednje modele
- Pri konceptualnem načrtovanju je zelo pomembno sodelovanje uporabnikov in interakcija z uporabniki. Uporabniki so nosilci znanja o poslovni domeni, so poznavalci semantike in običajno vedo, kaj hočejo, a to težko natančno izrazijo
- Konceptualno načrtovanje mora upoštevati tudi poslovna pravila (kot omejitve modela)

1.3 Tehnike konceptualnega načrtovanja..

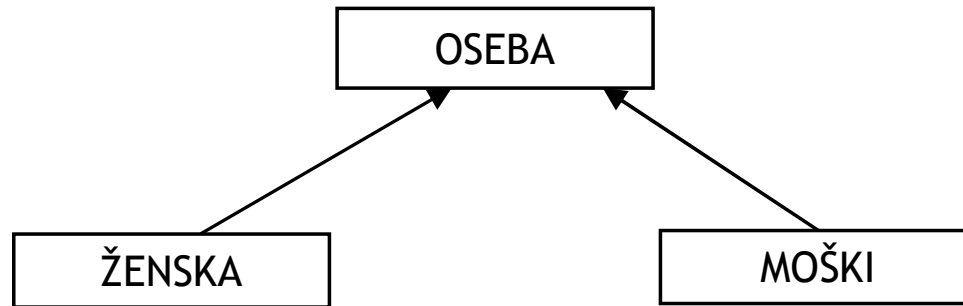
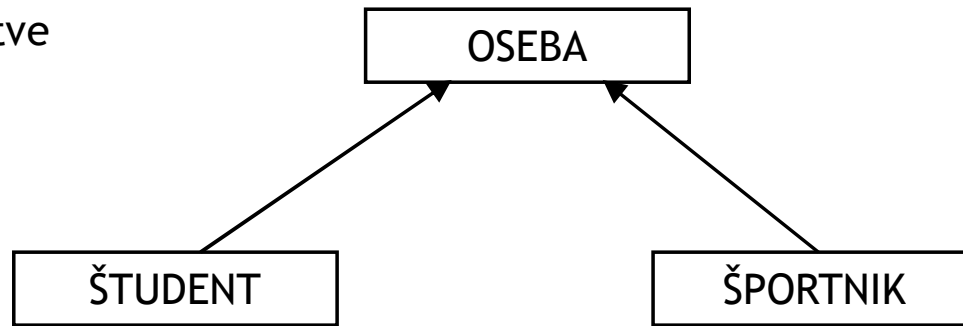
- Percepcija
- Abstrakcija
- Objekti, koncepti (tipi) in término
- Klasifikacija (objekti→koncepti)
- Ureditev hierarhij konceptov (tipov)
- Agregacija (sestavljanje konceptov)

1.3 Tehnike konceptualnega načrtovanja..

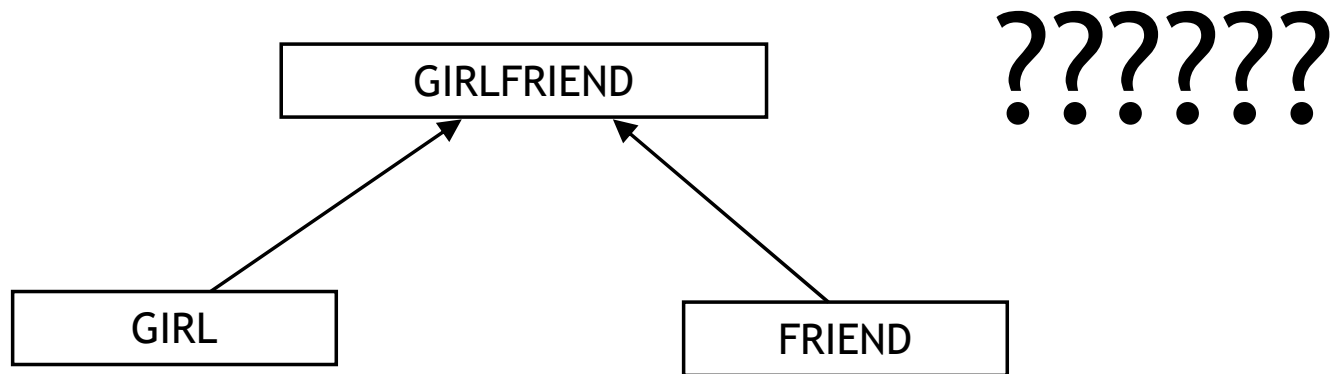


1.3 Tehnike konceptualnega načrtovanja..

Primeri ureditve tipov:
- pokrivanje?

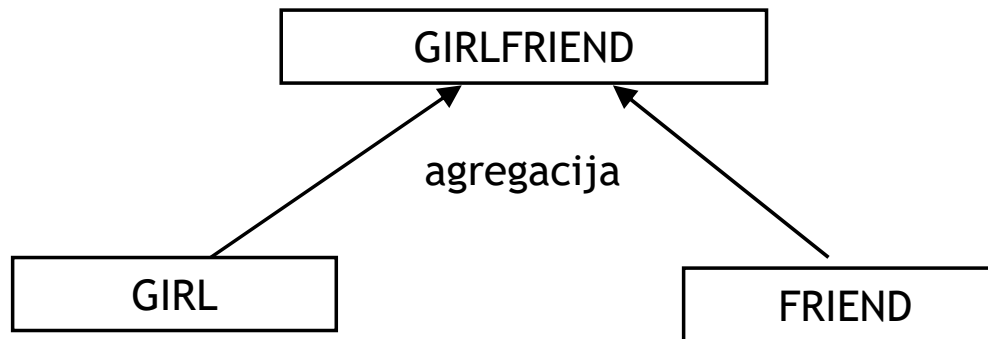


1.3 Tehnike konceptualnega načrtovanja..



1.3 Tehnike konceptualnega načrtovanja

- Preko agregacije definiramo nov – agregiran (sestavljen) tip na osnovi obstoječih tipov



- Kakšne je razlika z generalizacijo? Agregirani tip vsebuje VSE lastnosti agregiranih tipov in ne samo skupnih!
- Pomensko agregirani tip ponuja nekaj novega.

1.4 Lastnosti konceptualnega modela..

- **Namembnost konceptualnega modela**
 - Razumevanje problema (načrtovalec)
 - Vrednotenje razumevanja (uporabnik)
 - Pomoč pri načrtovanju in implementaciji (razvijalec)
- **Zahtevane lastnosti konceptualnega modela**
 - Izraznost (prikaz različnih konceptov),
 - Preprostost (enostaven za uporabo in razumevanje),
 - Minimalnost (vsak koncept predstavljen enolično),
 - Formalnost (natančen, nedvoumno definiran pomen konceptov)
 - Grafična popolnost (vsi koncepti razpoznavno grafično predstavljivi)
 - Berljivost (pregledna predstavitev gradnikov in celotnega modela).

1.5 Entitetni model..

- Najpogosteje uporabljana tehnika za predstavitev konceptualnih podatkovnih modelov sta entitetni model (model entiteta-razmerje) ter razredni diagram. Obravnavali bomo entitetni model.
- Nazivi, ki se uporabljajo:
 - Konceptualni podatkovni model
 - Podatkovni model
 - Entitetni model
 - ER model
- Razširjeni entitetni model

1.5 Entitetni model..

- Entitetni tip
- Atribut
- Razmerje
- Identifikator

1.5 Entitetni model..

Grafična predstavitev gradnikov entitetnega modela

- Originalni Chenovi diagrami

oznaka  za razmerja (relacije)

- Običajni ER diagrami: vranja noga (crow foot, IE)

oznaka  za nekatera razmerja (relacije)

- Chenovi diagrami so bolj izrazni, vendar kompleksnejši in dopuščajo različne interpretacije razmerij.

- UML (Unified Modeling Language)

-

1.5 Entitetni model..

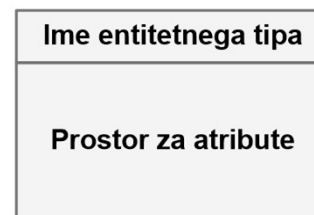
- Entitete so posamezne instance (primerki) tipov objektov iz poslovne domene: dogodki, predmeti, osebe, pravila, dejstva
- O entitetah obstaja določena predstava o tem:
 - kakšne lastnosti dejansko imajo
 - kakšne lastnosti jim moramo določiti (morajo imeti), da bodo izpolnjevale poslanstvo entitetnega modela
- Na osnovi predstave o tem in percepcije, lahko entitete klasificiramo v entitetne tipe: vse entitete, ki ustrezajo določeni predstavi, pripadajo posameznemu entitetnemu tipu.
- Primer: študenti

1.5 Entitetni model..

- Vsak trenutek pripada posameznemu entitetnemu tipu množica entitet tega entitetnega tipa, ki jo imenujemo entitetna množica
- Entitetna množica je časovno spremenljiva: entitete nastajajo, se spreminjajo in tudi izginjajo (izstopajo iz množice).
- Entitetna množica je v nekem trenutku lahko tudi prazna.
- Natančno moramo poznati pomen entitetnega tipa: kaj predstavljajo entitete, ki mu pripadajo

1.5 Entitetni model..

- V praksi se pogosto uporablja poenostavljen izraz entiteta, čeprav bi se moral uporabljati izraz entitetni tip



1.5 Entitetni model..

- Entitete imajo določene lastnosti, posamezne entitete (iz istega entitetnega tipa) se med seboj razlikujejo po njihovi vrednosti
- Le del entitetnih lastnosti je zanimiv oz. pomemben za opazovano poslovno domeno (abstrakcija)
- Lastnosti, ki so pomembne za opazovano poslovno domeno, vključimo v konceptualni model tako, da jih kot attribute določimo entitetnemu tipu

1.5 Entitetni model..

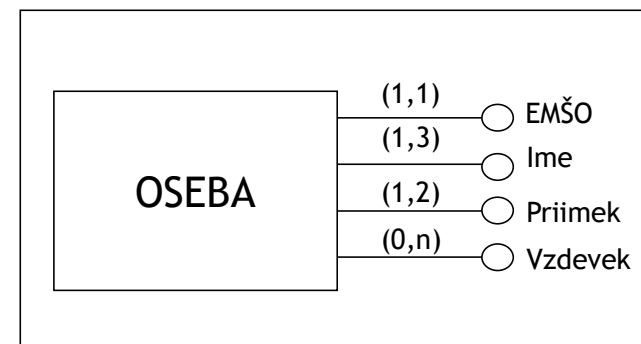
- Torej: z atributi formalno opišemo lastnosti entitet
- Govorimo o več vrstah lastnosti:
 - Entitetna imena: naziv, ime, opis
 - Prave entitetne lastnosti: višina, teža, cena, vrednost
 - Lastnosti, ki jih določimo za potrebe poslovnih procesov, poslovnih funkcij in poslovnih pravil: šifre, statusi, ...
- Atribut določimo za tisto lastnost, ki je za problemsko/poslovno domeno pomembna
- Vsak atribut ima določene lastnosti: kardinalnost (števnost), tip, dolžina

1.5 Entitetni model..

- Kardinalnost atributa omejimo z minimalno in maksimalno vrednostjo (min,max):

– Totalni atribut	(1,n), kjer je $n \geq 1$
– Parcialni atribut	(0,n), kjer je $n \geq 1$
– Enovrednostni atribut	(m,1), kjer je $m \in \{0,1\}$
– Večvrednostni atribut	(m,n), kjer je $m \in \{0,1\}$ in $n > 1$

- Minimalna števnost 0 pomeni, da je atribut lahko brez vrednosti (ni obvezen).
- Maksimalna števnost n pomeni, da atribut lahko zavzame poljubno število vrednosti.



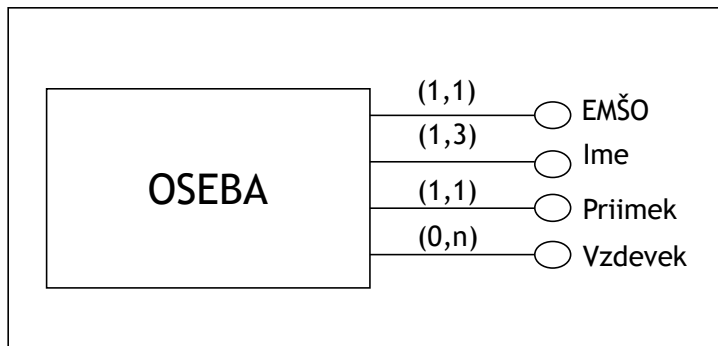
1.5 Entitetni model..

- Atribut pripada določenemu abstraktnemu tipu: numerični, znakovni, ...
- Za večino tipov je potrebno določiti tudi dolžino.

- V Chenovih ER diagramih imajo atributi lahko poljubno kardinalnost
- V običajnih ER diagramih imajo atributi kardinalnost vedno (0,1) ali (1,1)
- Večvrednosten atribut predstavimo z drugimi gradniki (nov entitetni tip in razmerje)

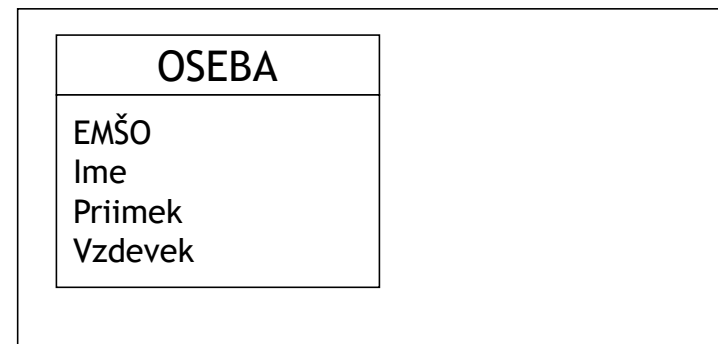
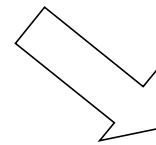
Atribut

1.5 Entitetni model..



Chenov diagram

Običajen ER diagram



1.5 Entitetni model..

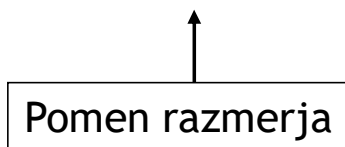
- Entitete niso svet zase, medsebojno se povezujejo preko razmerij (relacij)
- Razmerje ima določen pomen in števnost
- Predstavitev razmerja v modelu entiteta-razmerje je povezava
- Med opazovanim parom (v splošnem podmnožici) entitetnih tipov je lahko več razmerij: OSEBA, KRAJ – stalno bivališče, začasno bivališče
- Razmerij je lahko veliko, izberemo samo tista, katerih evidenco želimo voditi!

1.5 Entitetni model..

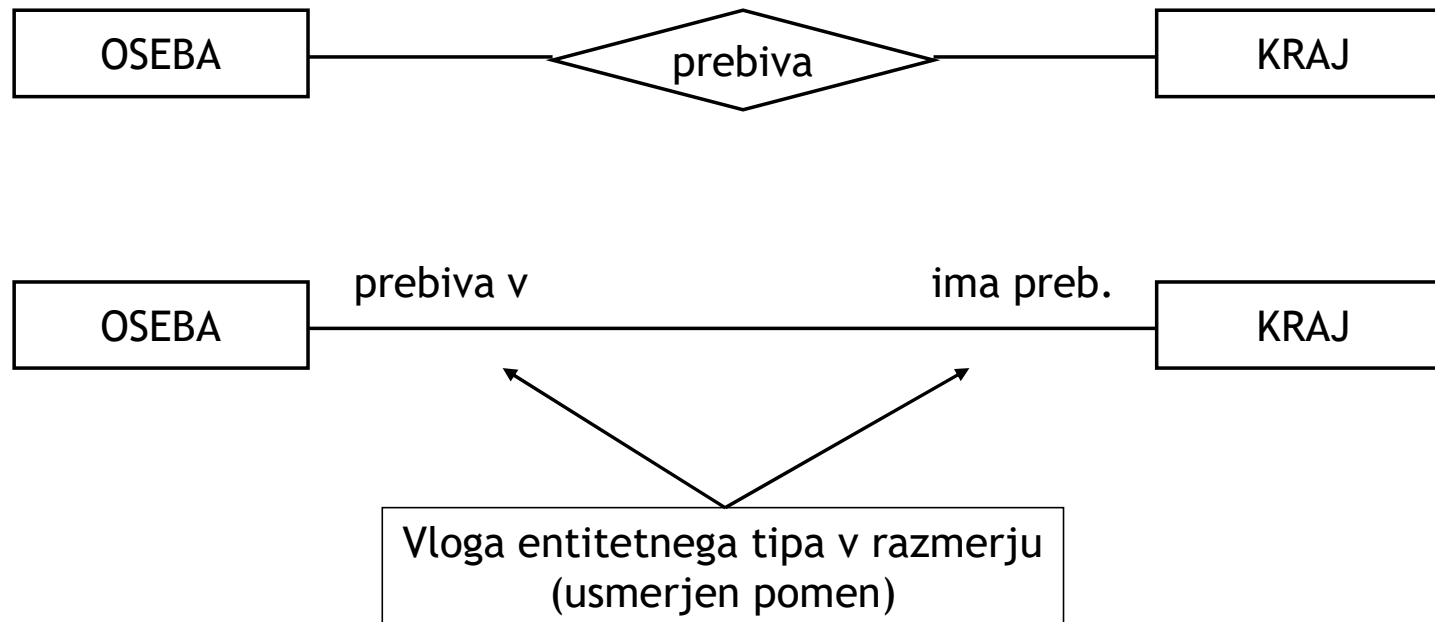
Chenov diagram



Običajen ER diagram



1.5 Entitetni model..

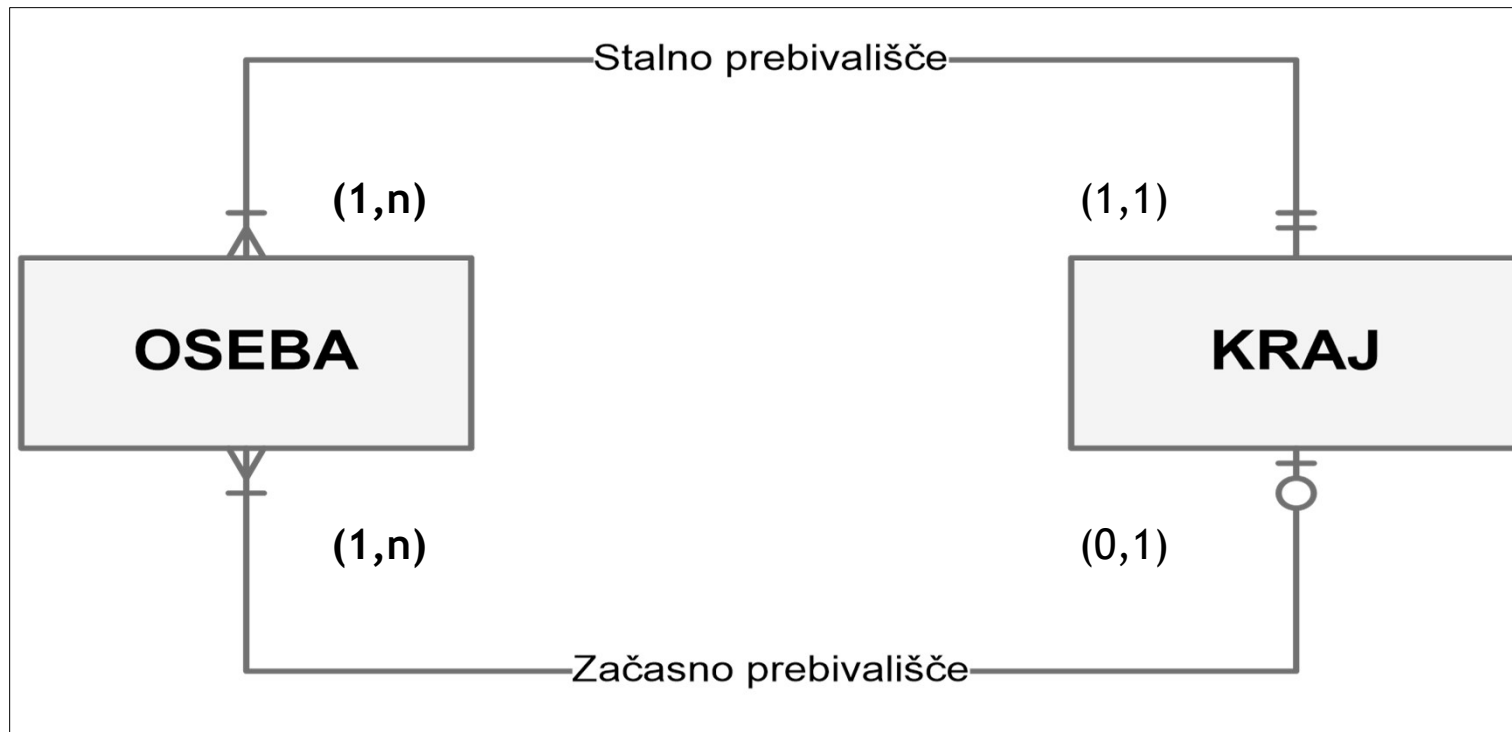


1.5 Entitetni model..

- Kardinalnost (števnost) predstavlja število entitet entitetnega tipa, ki so v razmerju glede na pomen razmerja. Minimalna in maksimalna števnost določata veljaven razpon števila entitet.
- Vsak entitetni tip ima svojo kardinalnost v razmerju glede na vlogo. Pomen razmerja med entitetnima tipoma OSEBA, KRAJ (na naslednji strani):
 - Ena (poljubna) oseba ima stalno bivališče v enem kraju (poštne enoti)
 - V enem (poljubnem) kraju (poštne enoti) ima stalno bivališče več oseb

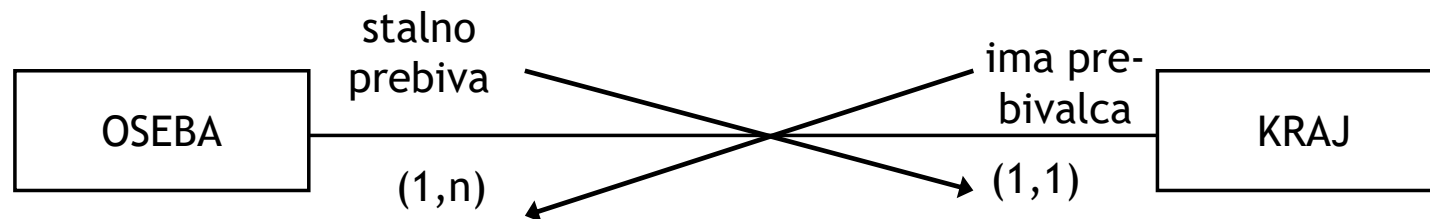
1.5 Entitetni model..


- Razmerji med entitetama OSEBA in KRAJ



1.5 Entitetni model..

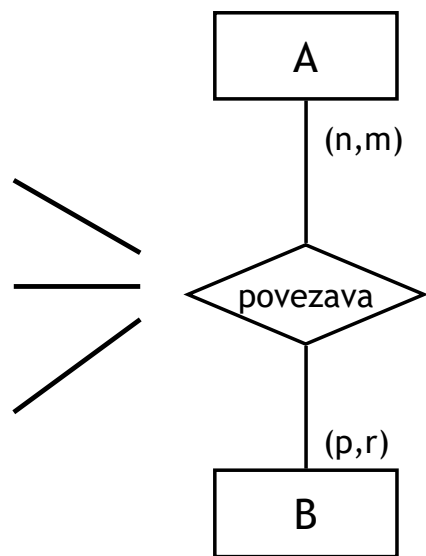
- Vloga in števnost v delnem razmerju



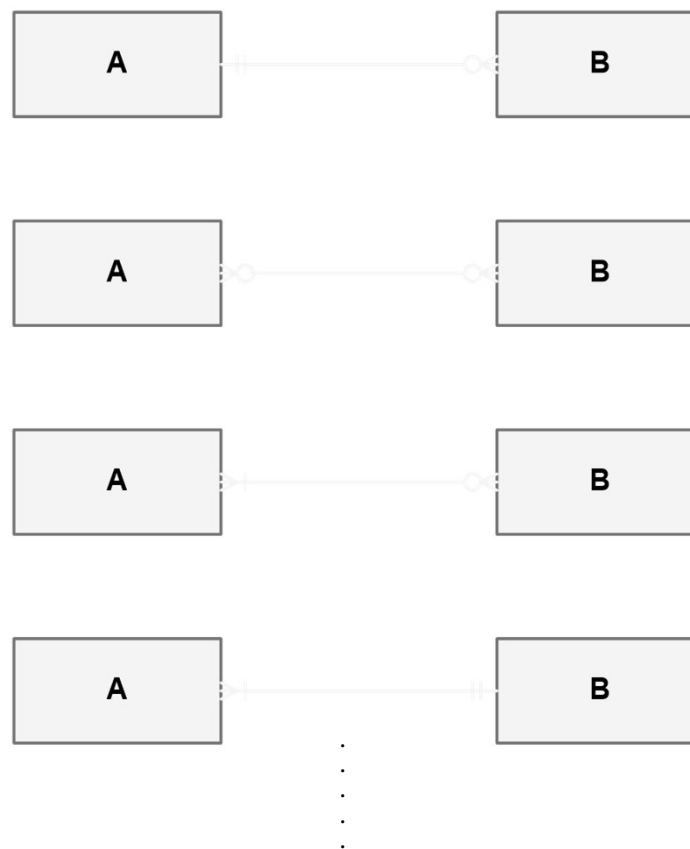
- Oznake za števnost na ponoru delnega razmerja:
 - 0: --o-- (krožec na povezavi)
 - 1: --|-- (pravokotna črta na povezavi)
 - več (m ali n): vranja noga  (crow foot)
 - Razpon določimo z dvema številka (minimalna, maksimalna) ali kombinacijo grafičnih oznak
 - (1,1) se označuje kot --|-- ali --||--
 - (0,1) se označuje kot --o-- ali --o|--

1.5 Entitetni model..

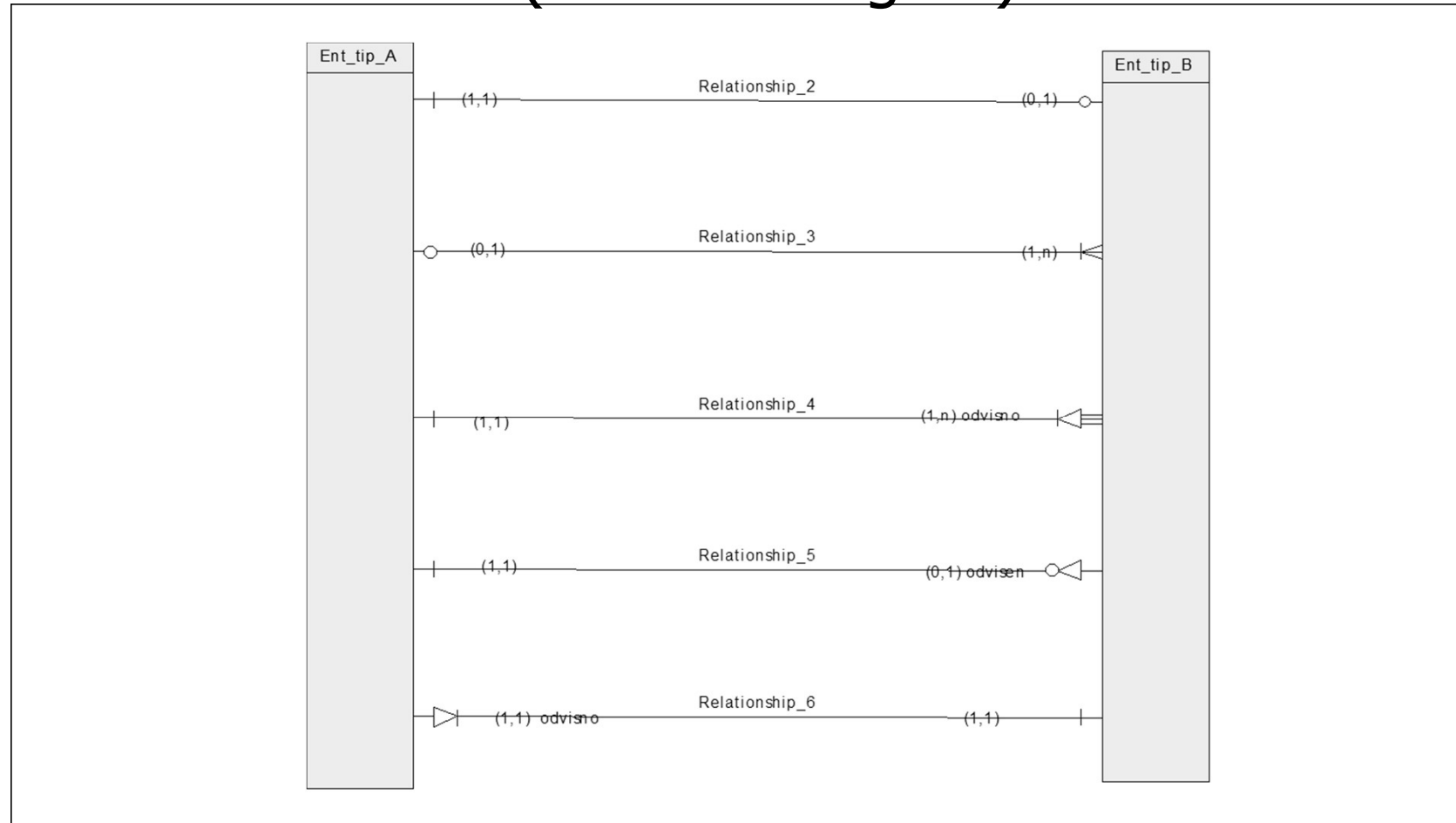
Chenov diagram (omogoča tudi več kot dvomestna razmerja)



Razmerje - kardinalnost



1.5 Entitetni model (Power Designer)



1.5 Entitetni model..

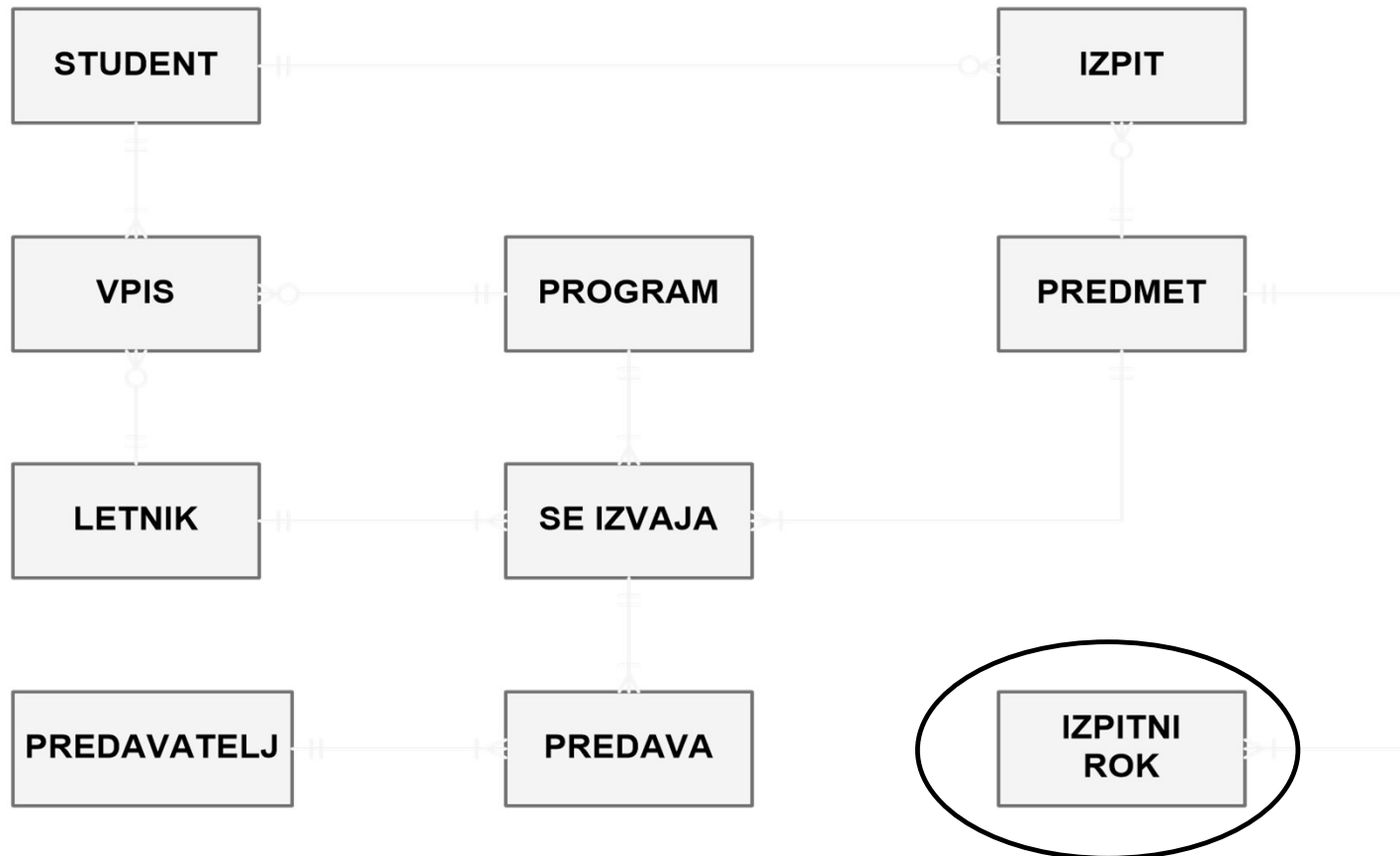
- Mandatnost pove, ali sta dve entiteti vedno v razmerju ali lahko tudi nista v razmerju: obvezno, neobvezno razmerje
- Mandatnost lahko obravnavamo pod okriljem števnosti, zaradi česar dodatno uvedemo števnost 0

1.5 Entitetni model..

- Razmerje tudi opisuje lastnost entitete
- Primer: OSEBA, KRAJ (poštna številka)
- Razmerje ima atributiven značaj
- Dilema: atribut (lastnost) ali razmerje?
 - Stalno (1,1) in začasno (0,1) prebivališče
 - Izpitni rok (3,3) ali (1,n)

Primer: eŠtudent

1.5 Entitetni model..



1.5 Entitetni model..

- Enolični identifikator entitete (krajše: entitetni identifikator ali samo identifikator) je podmnožica lastnosti entitetnega tipa (atributov in razmerij do drugih entitetnih tipov), ki enolično razlikujejo posamezne entitete znotraj entitetne množice
- Glede na to, ali tvorijo identifikator entitete le lastni atributi entitetnega tipa ali pa je v enoličnem identifikatorju tudi kakšno razmerje, ločimo med močnim entitetnim tipom in šibkim entitetnim tipom

1.5 Entitetni model..

- Imamo lahko več enoličnih identifikatorjev, vendar moramo enega izbrati – določiti
- Izbrani – določeni enolični identifikator je podlaga za (primarni) ključ v relacijskem modelu

Identifikator:

Abstrakten pojem

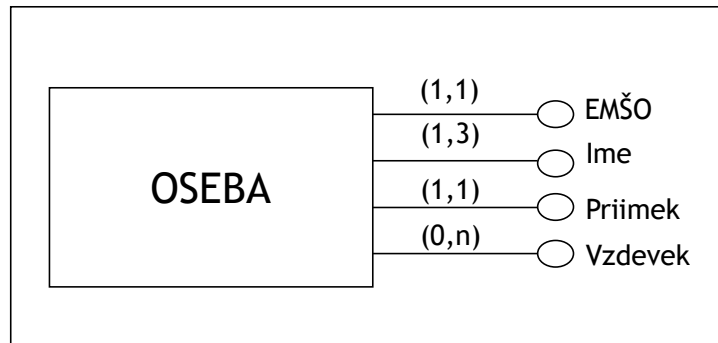
Ključ:

Relacijski model

- Identifikator se preslika v ključ relacijske sheme.

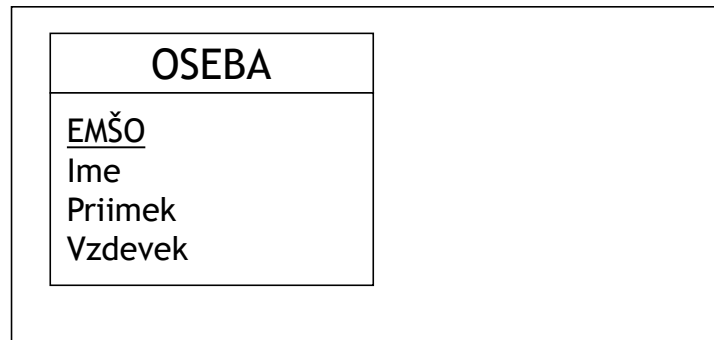
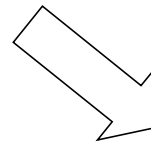
Enolični identifikator entitete

1.5 Entitetni model..

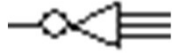


Chenov diagram
(pobarvamo piko ali
več pik, če gre za
sestavljen
identifikator)

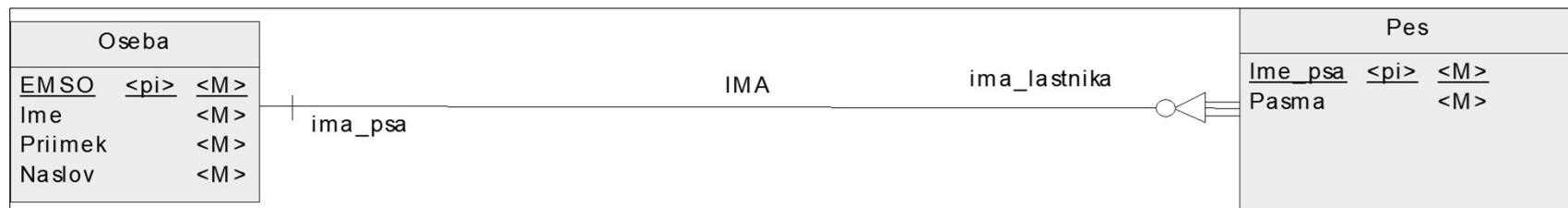
Običajen ER diagram
(podčrtamo atribut ali več
atributov, če gre za
sestavljen identifikator)




1.5 Entitetni model..

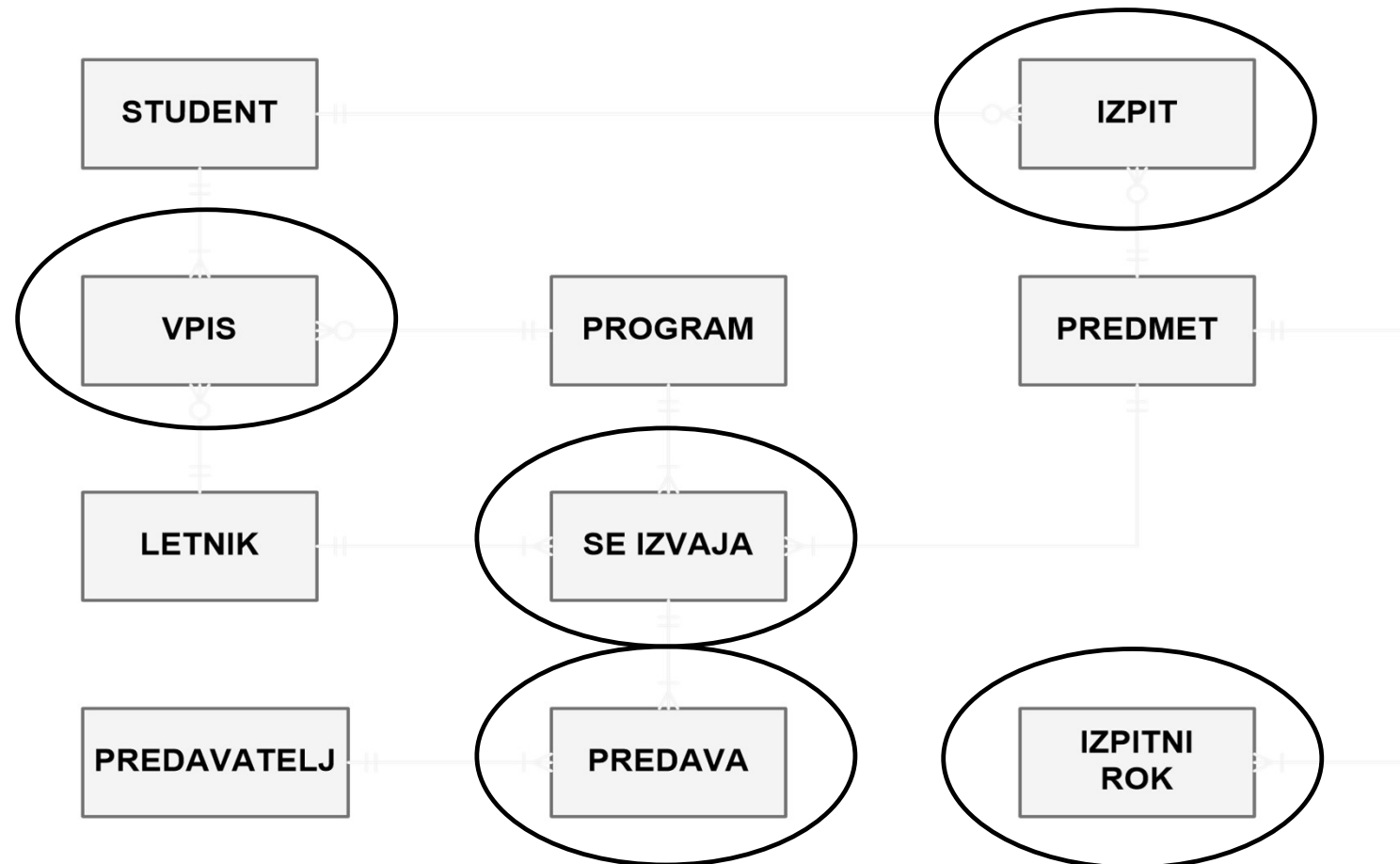
- Glede na atribute, ki sestavljajo identifikator, ločimo entitetne tipe na
 - Močne:
 - za identifikacijo objekta zadoščajo lastni atributi
 - Šibke:
 - za identifikacijo objekta lastni atributi ne zadoščajo
 - potrebujemo še nekaj **tujih** atributov
 - dobimo jih kot primarne ključe povezanih entitetnih tipov, od katerih je naš šibki entitetni tip **odvisen**.
 - odvisnost označimo s puščico (trikotnikom) na povezavi: 
 - Šibki entitetni tipi pogosto nastanejo po pretvorbi iz večmestnih (>2) ali večvrednostnih (m:n) razmerij, ter razmerij z atributi

1.5 Entitetni model..



- Trikotnik  označuje odvisnost (dependency) šibkega entitetnega tipa od močnega (v smetri puščice trikotnika). Kardinalnost: (0,n)
- Identifikator šibkega ent. tipa pes: Ime_psa in EMSO skupaj!
- Identifikator močnega entitetnega tipa Oseba: EMSO

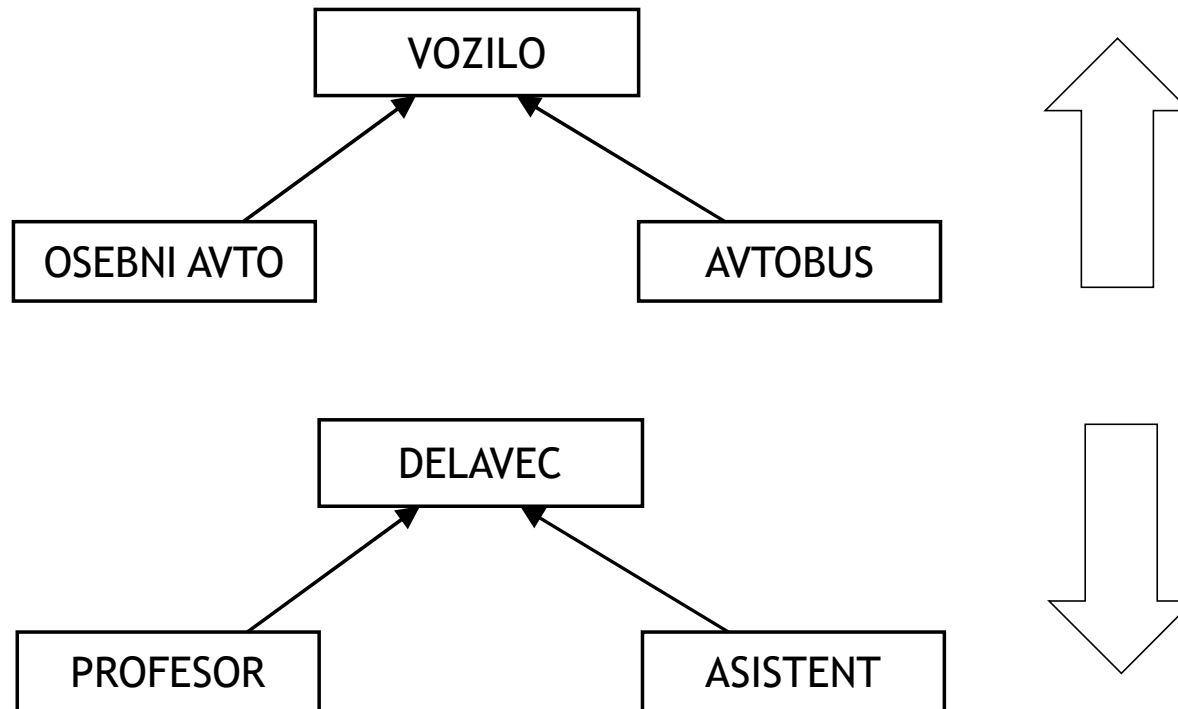
1.5 Entitetni model: identificirajte šibke entitetne tipe!



1.5 Razširjeni entitetni model..

- Razvrščanje sorodnih entitetnih tipov v hierarhije
- Izognemo se redundantnosti, povečamo semantiko
- Generalizacija
 - Imamo entitetna tipa B in C
 - Generaliziramo (posplošimo) ju v nadtip A
 - Pri tem se skupni atributi prenesejo na nadtip
- Specializacija:
 - Imamo entitetni tip A
 - Specializiramo ga v podtipa B in C
 - Podtipa ne vsebujeta skupnih atributov.

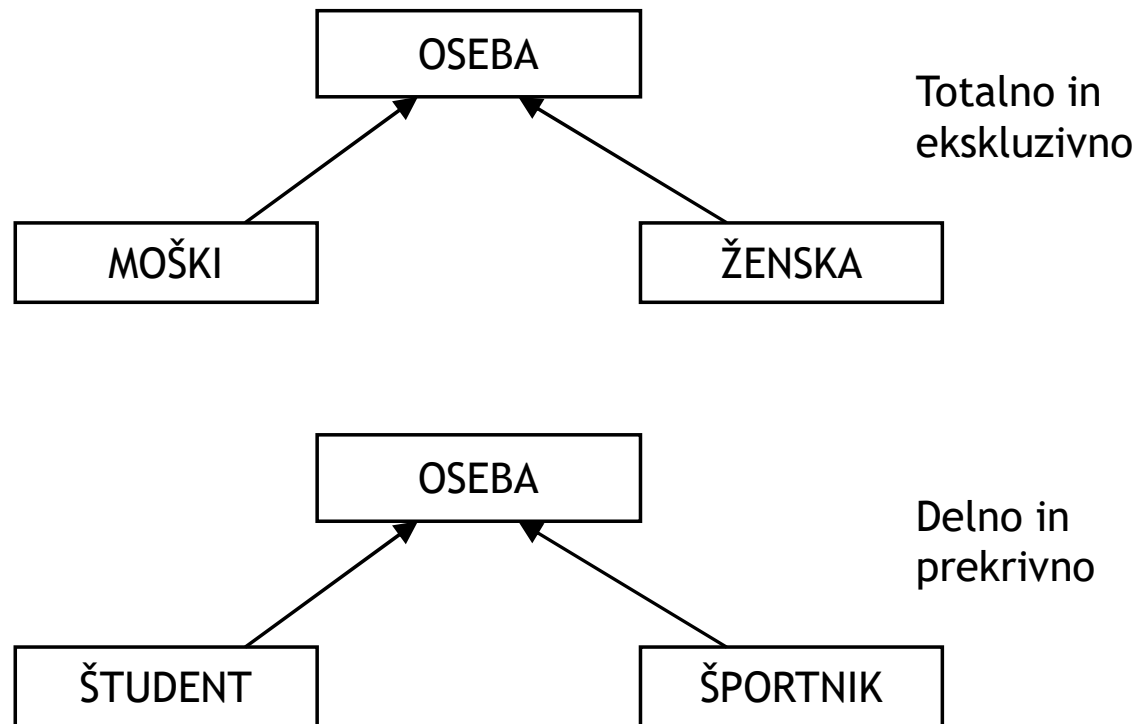
1.5 Razširjeni entitetni model..



1.5 Razširjeni entitetni model..

- Entitetni tip A s podtipoma B in C
- B in C pokrivata A totalno in ekskluzivno, če velja:
 $E_B \cup E_C = E_A$ in $E_B \cap E_C = \{\}$
- B in C pokrivata A totalno in prekrivno, če velja:
 $E_B \cup E_C = E_A$ in $E_B \cap E_C \neq \{\}$
- B in C pokrivata A delno in ekskluzivno, če velja:
 $E_B \cup E_C \subset E_A$ in $E_B \cap E_C = \{\}$
- B in C pokrivata A delno in prekrivno, če velja:
 $E_B \cup E_C \subset E_A$ in $E_B \cap E_C \neq \{\}$

1.5 Razširjeni entitetni model..



1.6 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 dojetje 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**

1.6 Metodologije in koncepti konceptualnega načrtovanja..

Strategija Z vrha navzdol

- Izhodiščna shema je entitetni tip, ki predstavlja shemo celotne poslovne domene (ali v nekem koraku sklop poslovne domene)
- Začetni vseobsegajoči 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

1.6 Metodologije in koncepti konceptualnega načrtovanja..

Strategija Od spodaj navzgor

- 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

1.6 Metodologije in koncepti konceptualnega načrtovanja..

Strategija Od znotraj navzven

- 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

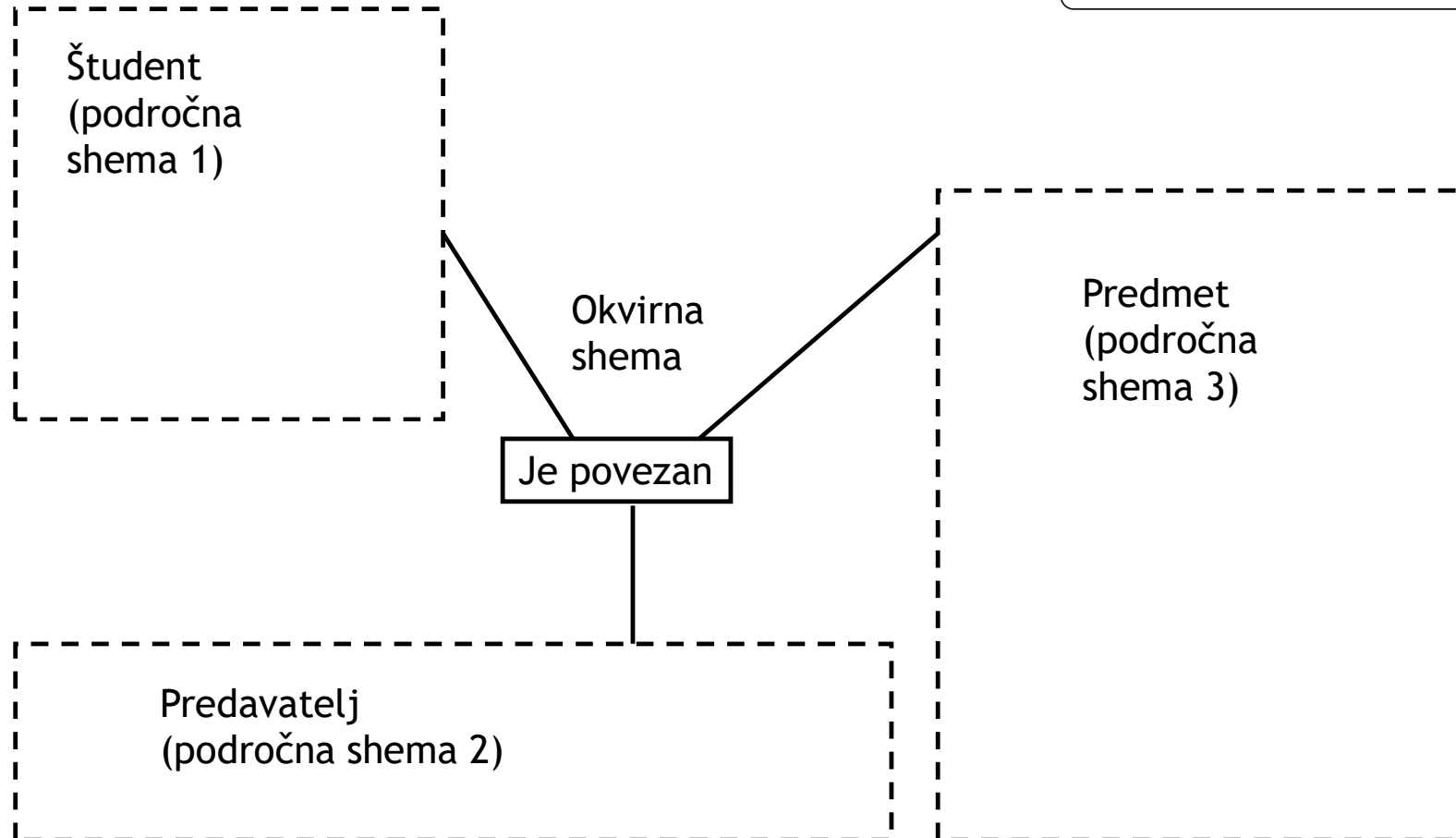
1.6 Metodologije in koncepti konceptualnega načrtovanja..

Strategija Po delih

- 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

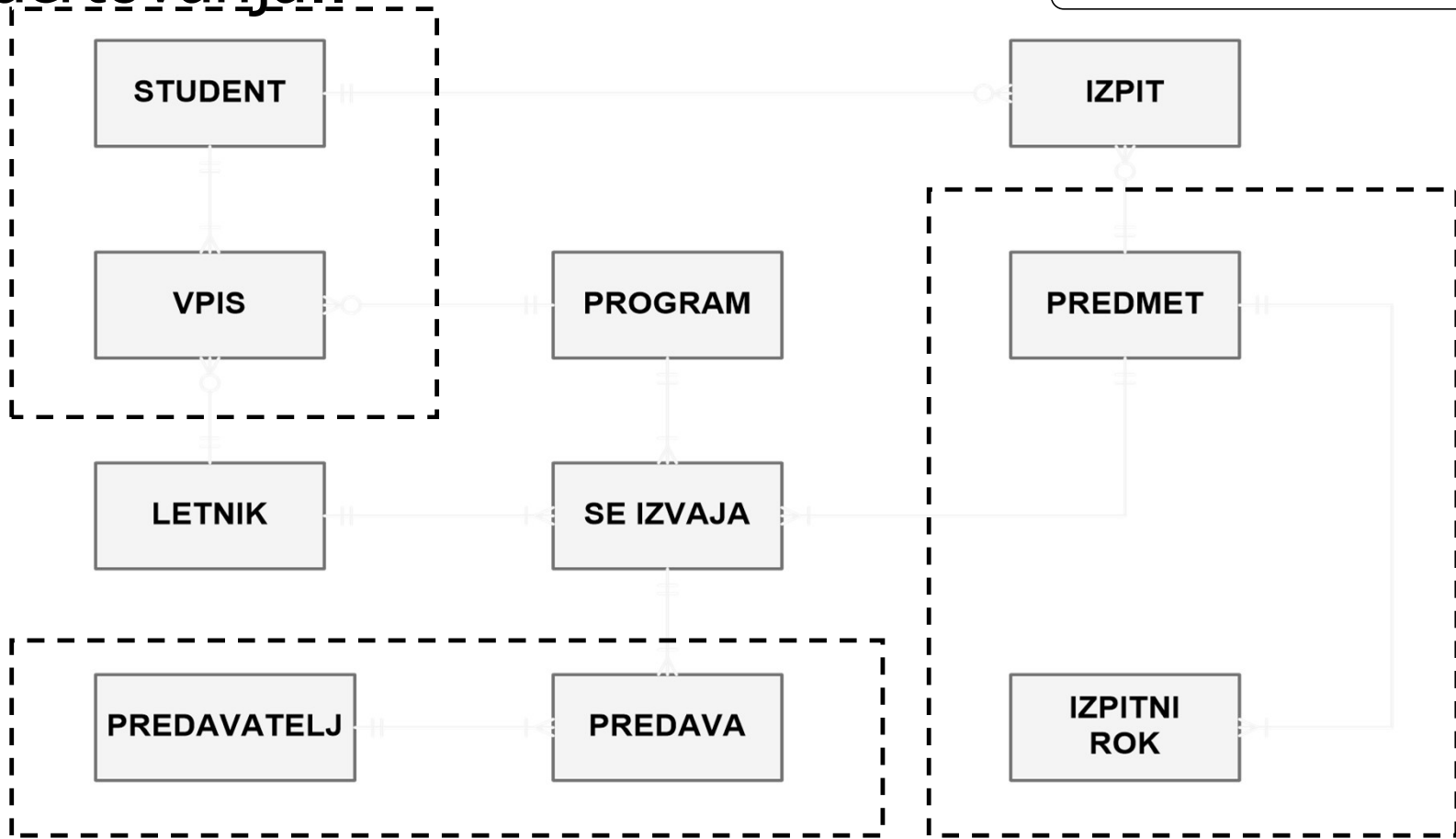
1.6 Načrtovanje po delih

Primer: eŠtudent



1.6 Metodologije in koncepti konceptualnega načrtovanja..

Primer: eŠtudent



1.6 Metodologije in koncepti konceptualnega načrtovanja..

Priporočila

- 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).

1.6 Metodologije in koncepti konceptualnega načrtovanja..

Priporočila - atribut ali nov entitetni tip in razmerje

- Načeloma naj entitetni tipi vsebujejo samo lastne attribute!
- 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

1.6 Metodologije in koncepti konceptualnega načrtovanja..

Priporočila - atribut ali nov entitetni tip in razmerje

- 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

1.6 Metodologije in koncepti konceptualnega načrtovanja..

Priporočila - odprava razmerij tipa M:N

- 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 attribute
- 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

1.6 Metodologije in koncepti konceptualnega načrtovanja..

Priporočila - odprava razmerij tipa M:N



1.6 Metodologije in koncepti konceptualnega načrtovanja..

Priporočila - Standardizacija imen

- 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**!

1.7 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

1.7 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"

1.7 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, attribute, kardinalnosti in mandatnosti (za attribute in razmerja)
- Pogovori (lahko) potekajo v več iteracijah, kjer iz iteracije v iteracijo vse bolj posegamo v podrobnosti

1.7 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

1.7 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 osebнем nivoju od dejanskih uporabnikov: pogovori ali izpolnjevanje vprašalnikov
 - Pogovori: prost opis problema, vključitev uporabnika v načrtovanje, sprotно 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.

1.7 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)

1.7 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

1.7 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 attribute
 - Zgradimo okvirno shemo
 - Zgradimo področne sheme
 - S pomočjo okvirne sheme povežemo področne sheme med seboj

1.7 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

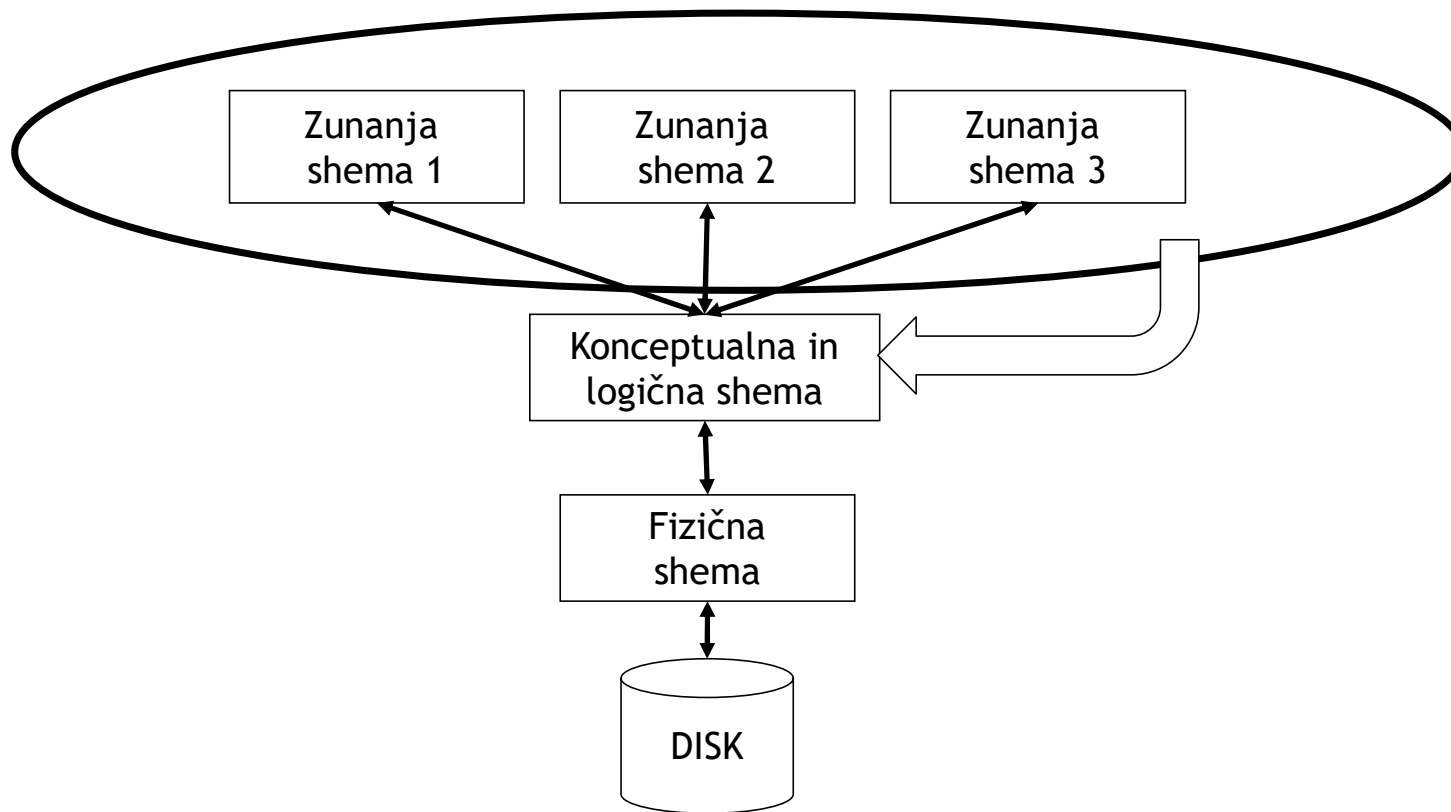
1.7 Načrtovanje zunanjih shem..

- 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

1.8 Združevanje zunanjih shem..

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

1.8 Združevanje zunanjih shem..



1.8 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
- Analiza neskladij je prvi korak združevanja podmodelov

} Pomenska neskladja

1.8 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

1.8 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



1.8 Združevanje zunanjih shem..

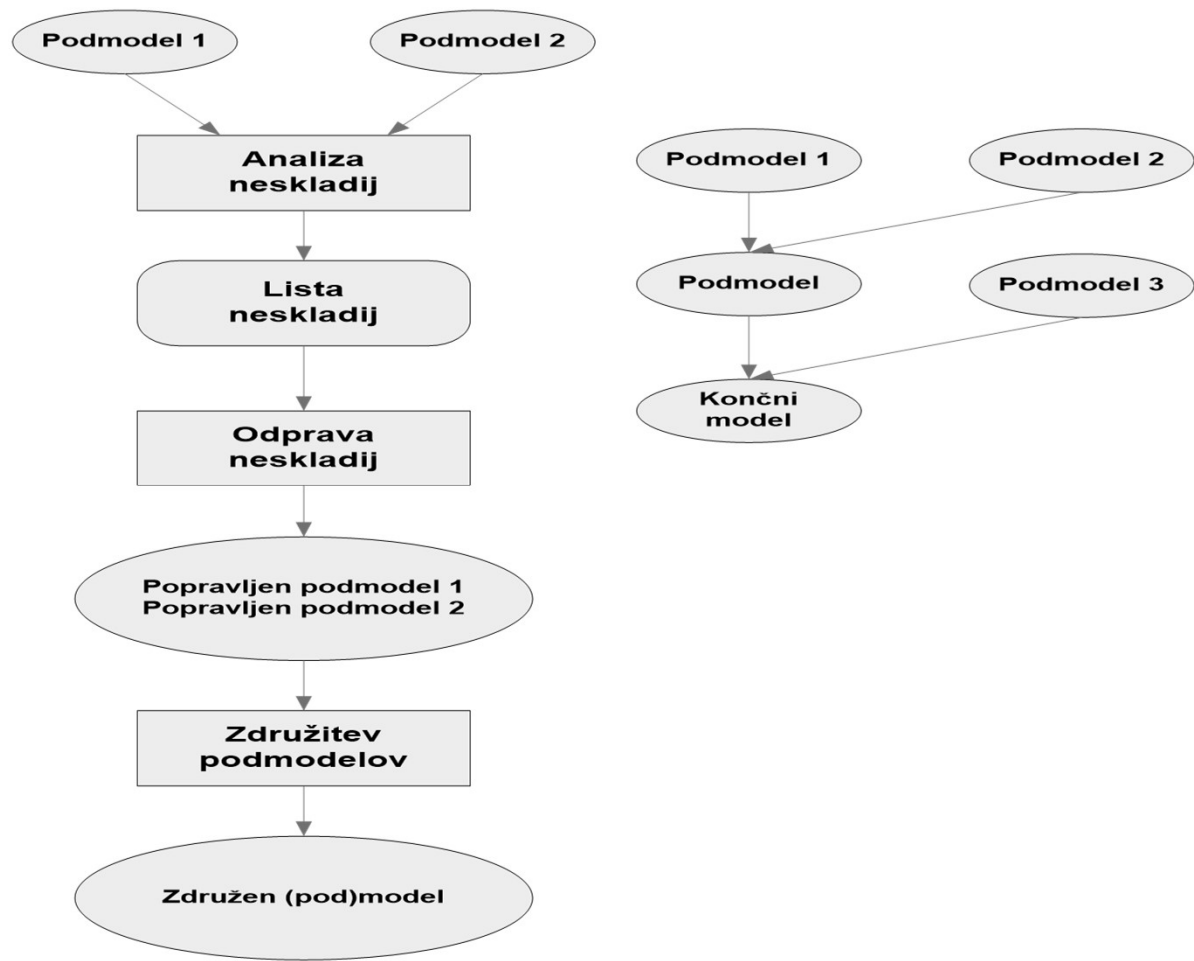
- Posledica dejstva, da posamezne dele modela lahko predstavimo na semantično ekvivalentne načine
- Pogosto gre za predstavitev iste lastnosti na eni strani z atributom, na drugi strani pa z entitetnim tipom in razmerjem

1.8 Združevanje zunanjih shem..

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

1.8 Združevanje zunanjih shem..

Združevanje zunanjih shem oz. podmodelov



1.8 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)

1.8 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

1.8 Združevanje zunanjih shem..

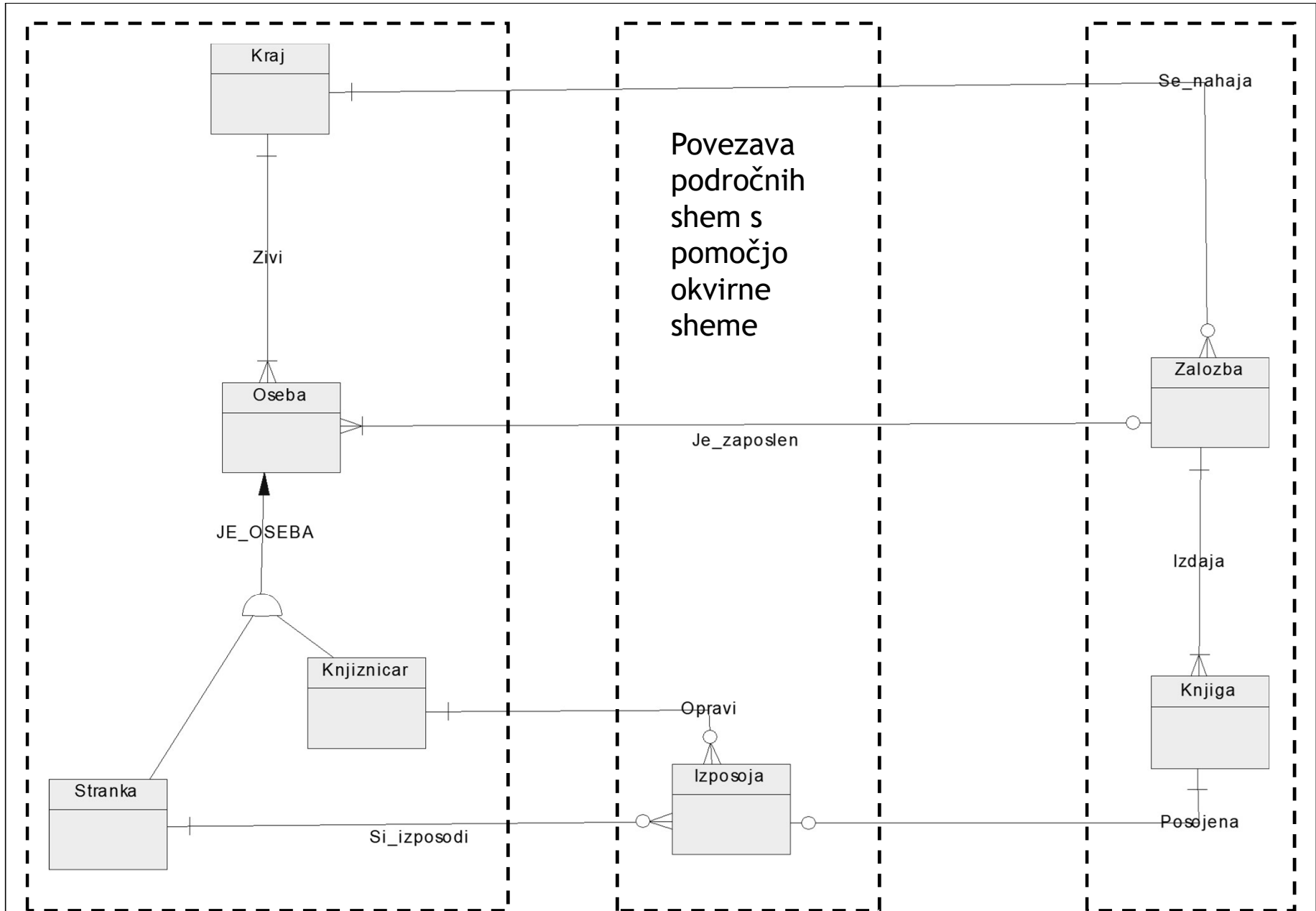
- Za entitetni tip kot vezni člen moramo združiti še attribute, kar izvedemo preko operacije unije
- Unija ne more biti povsem avtomatična. Tudi za združitev atributov je potrebno ugotoviti homonime in sinonime ter ustrezno ukrepati
- Entitetni tip vezni člen "prevzame" vsa razmerja, ki so bila pred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

1.8 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) attribute, ki jih le-ta podeduje od nadtipa

1.8 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 (doslej še neodkrito) razmerje med entitenim tipom prvega podmodela in entitetnim tipom drugega podmodela
- Podmodela lahko združimo preko na novo določenega razmerja



1.8 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

1.8 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 pod-modelov

Celovit pristop h konceptualnemu načrtovanju z orodji CASE (npr. Power Designer)

▪ Koraki konceptualnega načrtovanja:

- K1.1: Identificiraj entitetne tipe
 - K1.2: Identificiraj povezave
 - K1.3: Identificiraj in z entitetnimi tipi poveži attribute
 - 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 po primerni strategiji
- } Dodelav detajlov konceptualne sheme
- } Validacija konceptualne sheme

1.10 Kvaliteta konceptualnega podatkovnega 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*

1.10 Kvaliteta konceptualnega podatkovnega 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

1.10 Kvaliteta konceptualnega podatkovnega modela..

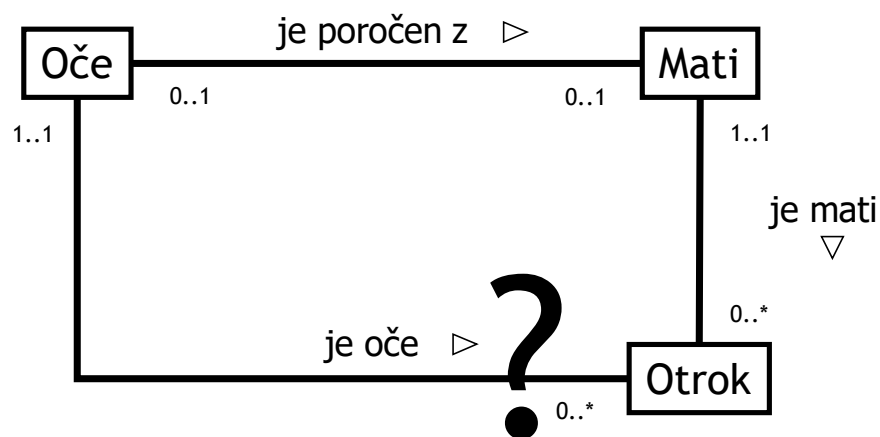
- Model je sintaktično pravilen, če je izdelan v skladu s sintaktičnimi pravili entitetnega modela (uporaba CASE orodja)
- Model je semantično pravilen, če vsi atributi in razmerja vseh entitetnih tipov ustrezajo poslovni domeni in njenim potrebam

1.10 Kvaliteta konceptualnega podatkovnega 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

1.10 Kvaliteta konceptualnega modela

- Preverimo, če v modelu obstajajo odvečni elementi:
 - Pregledamo povezava 1 – 1 (po možnosti združimo povezana entitetna tipa)
 - Odstranimo odvečne povezave (npr. tranzitivne)
 - Preverimo "časovni okvir" (semantika razmerja)



1.10 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

1.10 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)

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 transakcije 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 attribute, 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

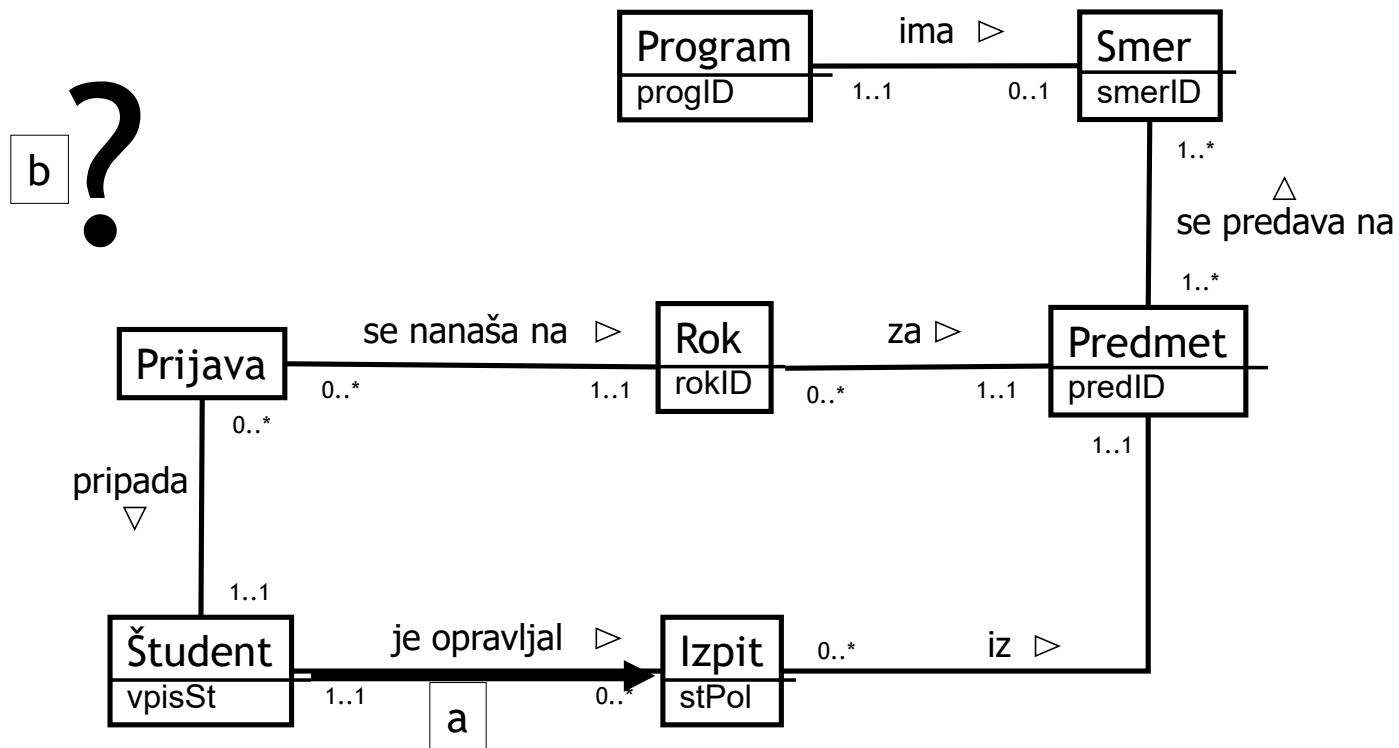
- Izpiši vse študente, ki so se vpisali v določen letnik, določene smeri, določenega programa
 - Izpiši vse predmete, ki jih je opravil določen študent
 - ...

Preverjanje transakcij

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

Primer preverjanja transakcij

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



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)