Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

**Igor Rožanc**

# Sistemska strukturirana metoda za analizo in načrtovanje

# (System Structured analysis and design Method - SSADM)

**Študijsko gradivo za interno uporabo pri**

**predmetu Razvoj programskih sistemov 2**

Ljubljana, 2007/08



## Kazalo

## Kazalo (nadaljevanje)





* - samo kratek oris



## SSADM: Uvod



**Namen:**

- ponuditi znanja s področja analize in načrtovanja poslovnih informacijskih sistemov – konkretno s spoznavanjem metodologij

**Dva pristopa:**

- plansko voden pristop : agilen pristop
- SSADM: (strukturirana) plansko vodena metodologija
- SCRUM, XP: agilni metodologiji (posebne prosojnice)

**Karakteristike strukturiranih metodologij:**

- strukturirajo projekt v majhne, dobro zaokrožene aktivnosti in določajo zaporedje in interakcijo teh aktivnosti
- uporabljajo diagramske in druge tehnike modeliranja, da bi dosegli bolj natančen strukturiran opis sistema, ki je razumljiv tako uporabnikom kot razvijalcem

## SSADM: Uvod



**Izbor plansko vodene metodologije:**

Za analizo in načrtovanje obstaja cela vrsta metodologij

Odločitev za SSADM: Structured Systems Analysis and Design Method (nastala 1982 na pobudo angleške vlade, trenutno verzija 4.2 iz 1995, v pripravi verzija 5)

**Prednosti:**

- ena najbolj popolnih in "zrelih" (mature, preizkušenih) metod
- zelo razširjena (v Vel. Britaniji je "standard" za razvoj aplikacij za potrebe vlade; anketa kaže, da jo uporablja 70 % organizacij, ki pri svojem delu uporabljajo strukturirane metode)
- se stalno dopolnjuje na podlagi izkušenj iz prakse (obstaja Technical Committee pri International SSADM user group)
- možnost pridobitve certifikata pri British Computer Society



## SSADM: Uvod



**Prednosti (nadaljevanje):**

- vključuje najboljše ideje iz ostalih metod (DFD iz SASD oziroma Gane&Sarson ter Yourdon&De Marco, podatkovno modeliranje z relacijsko analizo podatkov iz J. Martina, Entity Life Histories iz Jackson JSD & JSP): mimogrede spoznate glavne značilnost še nekaterih drugih metodologij
- temelji na usklajenem opisu načrtovanega sistema s treh vidikov:
  - **podatkovni vidik:** LDM Logical Data Model (entitete, razmerja)
  - **postopkovni vidik** oziroma pregled podatkovnih tokov: DFM Data Flow Model (kako se podatki pretakajo v sistem in iz njega in kako se transformirajo znotraj sistema)
  - **časovni vidik:** ELH Entity Life Histories (kako posamezni dogodki skozi čas spreminjajo podatke v sistemu)
- definira ustrezno zaporedje korakov, ki zagotavljajo medsebojno skladnost in navzkrižno kontrolo vseh 3 opisov

## SSADM: Uvod



**Slabosti:**

- včasih zahteva preveč podrobno opisovanje sistema (preveč birokratska)
- paralysis par analysis (paraliza zaradi analize)

**Rešitev:**

- dobro poznavanje metodologije
- izkušnje
- prilagajanje metodologije konkretnemu projektu

**Prilagajanje je možno šele, ko metodologijo v celoti poznamo: opisali bomo metodologijo tako, kot je, šele nato bomo govorili o možnostih za prilagajanje**



## SSADM: Splošen opis



**SSADM in življenjski cikel sistema**

**Življenjski cikel (po Goodlandu):**

- Planiranje IS (Information systems planning)
- Vzpostavitev projekta (Project initiation)
- Študija izvedljivosti (Feasibility study)
- Sistemska analiza (Systems analysis)
- Logični (poslovni) načrt (Business systems design)
- Fizični načrt (Physical design)
- Izdelava (Construction): Programiranje
- Prehod na nov sistem (Transistion)
- Obratovanje (Production)
- Vzdrževanje (Maintenance and Review)

## SSADM: Splošen opis



**Strateško planiranje IS:**

- v organizacijah, ki se zavedajo pomena IS
- možna uporaba posebnih metod
- nekatere od tehnik, ki jih uporablja SSADM, so uporabne:
  - Definicija zahtev (Requirements Definition)
  - Diagrami podatkovnih tokov (DFD)
  - Logične podatkovne strukture (LDS)

**Zaključek: to področje je delno pokrito s SSADM**

**Vzpostavitev projekta:**

- pogoji za izvedbo, okvirni obseg, določitev delovne skupine, plan

**SSADM vsebuje samo nekaj priporočil za to fazo**



## SSADM: Splošen opis



**Študija izvedljivosti:**

- Tri vprašanja:
  - ali je projekt tehnično izvedljiv ?
  - ali je finančno upravičen ?
  - kakšne so ostale posledice (socialne, kadrovske, ...) ?
- v zadnjem času so te študije manj popularne
- delno se prenesejo v strateško planiranje
- pogosto je projekt nujno potreben (must have), zato vprašanje izvedljivosti odpade
- **SSADM vsebuje podrobna navodila za to fazo, a je mi ne bomo posebej obravnavali (ti. faza 0)**

## SSADM: Splošen opis



**Sistemska analiza:**

- poudarek na analizi obstoječega stanja z namenom definirati zahteve bodočega sistema
- **SSADM to pokriva v celoti v fazah 1 in 2**

**Logični načrt novega sistema:**

- izdelava specifikacije zahtev za nov sistem
- vrednotenje tehničnih rešitev, ki pridejo v poštev za realizacijo, izbor najustreznejše
- logični načrt, ki na ne-tehničen način opisuje delovanje novega sistema
- **SSADM to pokriva v celoti v fazah 2,3,4 in 5**



## SSADM: Splošen opis



**Fizični načrt:**

- konverzija logičnega načrta v obliko, ki se sklada z izbrano strojno in programsko opremo
- izdelava fizičnega načrta podatkovne baze
- specifikacija programov
- opis potrebnih operacij in ročnih postopkov
- **SSADM to pokriva v celoti v fazi 6**

**Izdelava:**

- plan izgradnje sistema (faza 4)
- plan testiranja (faza 6)
- možnost uporabe jezikov 4. generacije v fazah 5 in 6
- prototipiranje v fazi 3
- **SSADM samo delno pokritje**

## SSADM: Splošen opis



**Prehod na nov sistem:**

- **samo delno pokritje: plan prehoda v fazi 4**

**Obratovanje:**

- **ni pokrito**

**Vzdrževanje:**

- odprava morebitnih napak
- prilagajanje novim verzijam SW in HW
- izboljšave (dodajanje novih funkcij)
- pregledi (reviews): ali sistem izpolnjuje zahteve
- **SSADM delno pokriva: obstajajo priporočila za vzdrževanje, ki jih ne bomo obravnavali**



## SSADM: Splošen opis

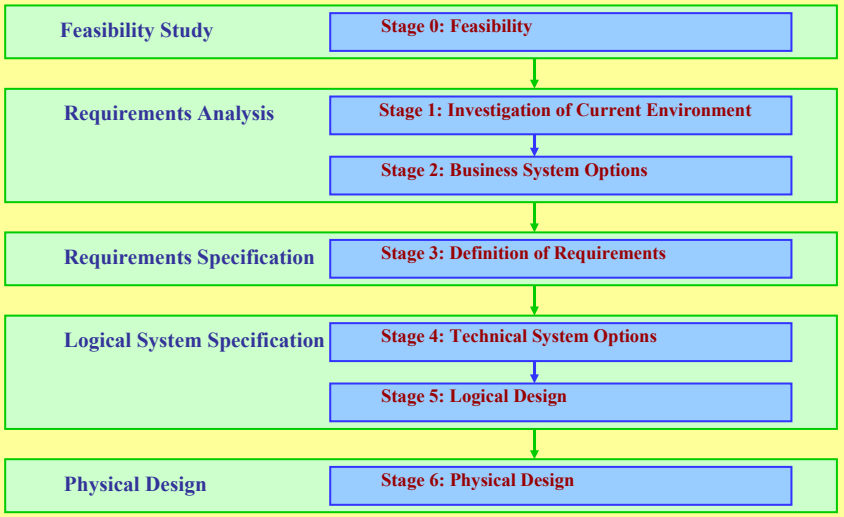


**Struktura SSADM**

- 5 modulov
- moduli so razdeljeni na faze (angl. stages)
- večina modulov vsebuje eno fazo, nekateri dve
- moduli so oštevilčeni od 0 do 6
- vsaka faza se nadalje deli na korake (angl. steps)
- koraki so označeni z trimestnimi številkami (XY0), kjer prva cifra (X) pomeni oznako modula, druga cifra (Y) pa zaporedno številko koraka v modulu
- za vsak korak so predpisani inputi (vhodi), outputi (izhodi) in naloge (angl. tasks), ki jih je treba izvršiti
- koraki se izvajajo zaporedoma ali istočasno
- nekatere korake lahko v celoti izpustimo

## SSADM: Splošen opis









## SSADM: Splošen opis



## SSADM: Splošen opis



**Značilnosti:**

- najprej proučujemo obstoječi sistem, da bi razumeli okolje, v katerem bo moral delovati nov sistem
- na osnovi obstoječega sistema izdelamo specifikacijo novega sistema, ki pa ni omejena s trenutno implementacijo
- specifikacijo zahtev definiramo tako podrobno, da lahko formuliramo različne tehnične opcije
- šele ko je načrt podrobno razdelan na logičnem nivoju, se ukvarjamo z implementacijo
- konverzija iz logičnega načrta v fizični poteka s pomočjo enostavnih pravil; dobljen načrt nato še optimiziramo



## SSADM: Splošen opis



**Pregled strukturiranih tehnik, ki jih uporablja SSADM**

Tehnike: kako se izvedejo posamezne naloge

**Delitev:**

- diagramske tehnike
- nediagramske tehnike (in postopki)

**Diagramske tehnike:**

- modeliranje podatkovnih tokov (DFD)
- logično modeliranje podatkov (LDS)
- modeliranje dogodkov nad posameznimi entitetami (ELH - Entity Life Histories)
- načrtovanje dialogov
- načrtovanje postopkov nad logično podatkovno bazo

## SSADM: Splošen opis



**Nediagramske tehnike:**

- definicija zahtev
- definicija funkcij
- relacijska analiza podatkov
- prototipiranje specifikacij
- fizično načrtovanje podatkovne baze
- specifikacija fizičnih procesov





## SSADM: Splošen opis



**Dokumentacija**

- Za diagramske tehnike obstajajo standardi (točno določena notacija), ki določajo simbole, izgled in vsebino izdelane dokumentacije
- Za ostale strukturirane informacije (npr. opis entitet, opis atributov, opis zahtev) obstajajo predlogi ti. "standardnih obrazcev", ki si jih specifični projekti lahko prilagodijo za svoje potrebe
- Včasih uporaba matrik: matrika entitet za določitev razmerij, matrika entiteta/dogodek kot izhodišče za Entity Life History diagrame (diagrame, ki ponazarjajo različna stanja v življenju entitete)
- Nestrukturirane informacije: format poročil in njihova struktura ni natančno predpisana; vanje je treba vkomponirati ostalo SSADM dokumentacijo
- Dokumentacija pogosto pogojena s CASE orodjem

## SSADM: Splošen opis



**TRIJE POGLEDI NA SISTEM**

**1) logični podatkovni model (LDM):**

kateri podatki se hranijo in kako so med sabo povezani

**2) model podatkovnih tokov (DFM):**

kako se informacije "pretakajo" po sistemu, vključno s procesi, ki jih preoblikujejo

**3) diagrami sprememb stanja entitet (ELH):**

kako se informacije spreminjajo v času svojega življenja



## SSADM: Logični podatkovni model



**Splošno**

- Zelo popularna tehnika, uporabljena v številnih metodologijah (npr. James Martin: Information Engineering), znana tudi iz podatkovnih baz
- Logični podatkovni model sistema prikazuje entitete in njihova razmerja

**a) Entiteta**

**Definicija entitete po SSADM (inženirska):** *An entity is something of significance to the system about which information is to be held.*

*Entiteta je nekaj, kar je pomembno za sistem, o katerem hranimo podatke.*

**Primeri informacijskih sistemov in pripadajočih entitet:**

- Bolnišnični IS: pacienti, oddelki, bolezni, recepti, zdravila
- Knjižnični IS: knjige, člani, izposoje, rezervacije, opomini
- Študijski IS: študenti, učitelji, predmeti, izpiti

## SSADM: Logični podatkovni model



**Entitetni tip : posamezne/konkretne entitete (type, occurence)**

Pacient: entitetni tip

Novak Janez, Kralj Lev, Kovač Miha: konkretne entitete

**Predstavitev entitet:**

- v pravokotniku
- ime entitete (tj. entitetnega tipa) v ednini z veliko začetnico




## SSADM: Logični podatkovni model



**b) Atribut**

**Definicija:**

*An attribute is the smallest discrete component of the system information that is meaningful.*

*Atribut je najmanjša diskretna komponenta informacije o sistemu, ki ima pomen.*

Drugače povedano:

- lastnosti posameznih entitet opišemo z atributi
- obravnavamo samo tiste atribute, ki so pomembni za sistem (v realnem svetu lahko nastopajo tudi drugi atributi)

**Sinonimi za atribut**: podatkovni element (data item, logical data item, data element), polje (field)

**Ločiti med atributnim tipom in vrednostjo atributa.**

**V nadaljevanju: "atribut" pomeni "atributni tip"**

## SSADM: Logični podatkovni model



**Primarni ključ:** *atribut (ali kombinacija atributov), katerih vrednosti enolično določajo neko konkretno entiteto*

**Primeri:** vpisna številka, številka bančnega računa, davčna številka

**Sestavljen ključ:** če ima vsaka bančna podružnica svoj sistem številčenja, je primarni ključ sestavljen iz številke podružnice in številke računa

**c) Razmerja**

Konkretne entitete v realnem svetu so v določenem razmerju z drugimi entitetami iz realnega sveta; npr. Janez Novak:

- ima odprt bančni račun
- je pacient na oddelku 10 v neki bolnici
- je študent na neki fakulteti
- si izposoja knjige v neki knjižnici

**Upoštevamo samo razmerja, ki so pomembna za sistem, ki ga načrtujemo**



## SSADM: Logični podatkovni model

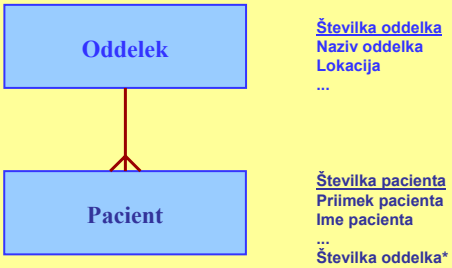


**Definicija:** *Razmerje je za sistem pomembna povezava med dvema entitetama.*

Pomen razmerij za dostop do posameznih entitet: razmerje omogoča, da za neko dejansko entiteto najdemo vse tiste entitete, ki so z njo povezane.

**Primer:** Razmerje Pacient:Oddelek

- za nekega pacienta lahko določimo oddelek, kjer se zdravi
- za nek oddelek lahko dobimo vse paciente, ki se tam zdravijo



## SSADM: Logični podatkovni model



**Realizacija dostopa:** preko skupnih atributov (v našem primeru Številka oddelka)

- v Pacient je Številka oddelka tuj ključ
- v Oddelek je Številka oddelka primarni ključ

**Tuj ključ:** *atribut, ki je v neki drugi entiteti primarni ključ*

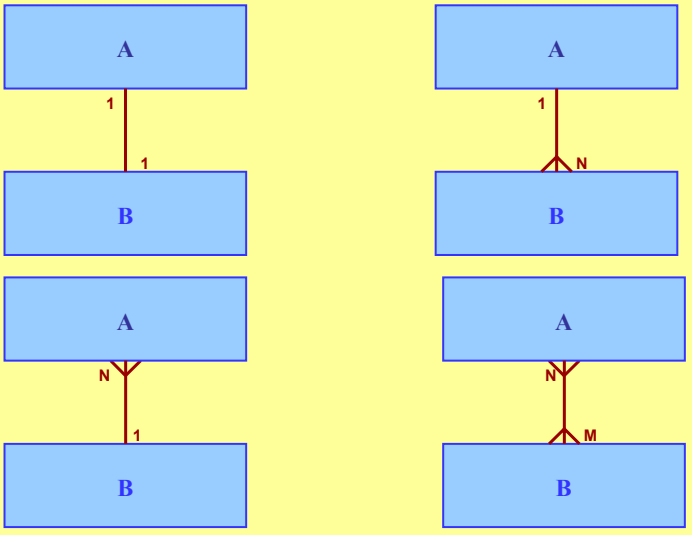
**Vrste razmerij:**

| | |
|---|---|
| **1:1** | Entiteta tipa A je povezana z eno samo entiteto tipa B |
| **1:N** | Entiteta tipa A je povezana z več entitetami tipa B |
| **N:1** | Več entiteta tipa A je povezanih z eno samo entiteto tipa B |
| **M:N** | Več entitet tipa A je povezanih z več entitetami tipa B |




## SSADM: Logični podatkovni model





## SSADM: Logični podatkovni model



**Primeri:**

Študent in Izpit (1:N)

- Vsak študent lahko opravi enega ali več izpitov.
- Izpit mora biti opravljen s strani enega in samo enega študenta.

Naročilo in Izdelek (M:N)

- Naročilo mora vsebovati 1 ali več izdelkov.
- Izdelek je lahko vsebovan v 1 ali več naročilih.

Pošiljka in Reklamacija (1:1)

- Vsaki pošiljki lahko pripada samo ena reklamacija
- Reklamacija se nanaša na eno samo pošiljko



## SSADM: Logični podatkovni model



**Opcijska in obvezna razmerja:**

- Razmerje med dvema entitetama je **opcijsko**, če ena dejanska entiteta lahko obstaja, ne da bi bila povezana z drugo (ne da bi obstajalo razmerje z drugo)
- Razmerje med dvema entitetama je **obvezno**, če ena dejanska entiteta ne more obstajati, ne da bi bila povezana z drugo.

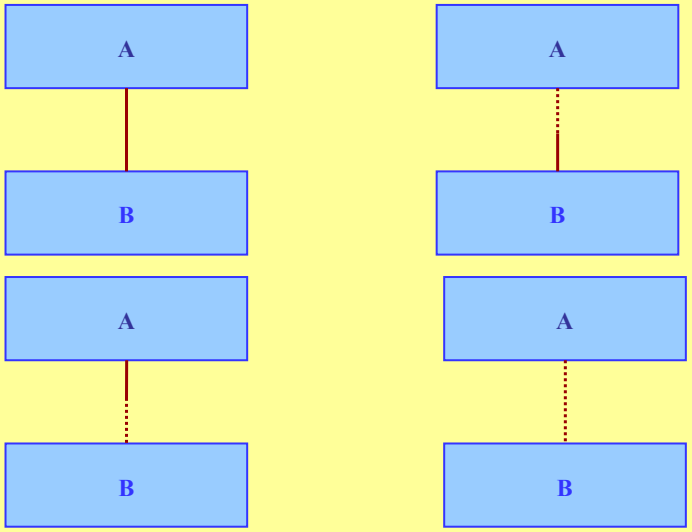
**Pravilo: črtkana črta je pri entiteti, ki lahko obstaja samostojno (brez povezave z drugo)**

Razmerje je opcijsko na:

- nobeni strani
- eni strani
- obeh straneh

## SSADM: Logični podatkovni model









## SSADM: Logični podatkovni model



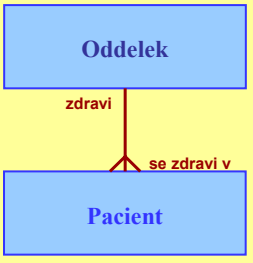
**Poimenovanje razmerij:**

- vsako razmerje poimenujemo na obeh koncih
- izberemo primeren povedek
- ta pomaga formirati smiseln stavek, s katerim lahko opišemo razmerje

**Primer: razmerje Oddelek:Pacient**

*Each ward must be treating one or more patients.*
*Each patient must be treated in one and only one ward.*

*Vsak oddelek mora zdraviti enega ali več pacientov.*
*Vsak pacient se mora zdraviti na enem in samo enem oddelku.*



## SSADM: Logični podatkovni model



**Splošna oblika stavka:**

| each | vsak(a) |
|---|---|
| *<subject entity>* | *<entiteta osebek>* |
| must be/may be | mora/lahko |
| *<link phrase closest to subject entity>* | *<povedek pri osebku>* |
| one and only one/ one or more | enega in samo enega/ enega ali več |
| *<object name entity>* | *<entiteta predmet>* |

**Izbor povezovalne fraze (povedka):**

- čim bolj smiseln in določen,
- ne splošen in dvoumen

**Dobro:** se zdravi, je avtor, je imetnik, si izposodi, izposoja, je zaposlen, zaposluje

**Slabo:** je povezan z, pripada, je del



## SSADM: Logični podatkovni model



**Prednosti poimenovanja razmerij:**

- olajša predstavitev modela uporabniku
- pomaga analitiku pri boljšem razumevanju poslovnega okolja (vpeljuje strogost, ki zagotavlja, da je korektno interpretiral poslovno okolje)

**Razmerja odražajo organizacijska pravila, značilna za poslovno okolje, ki ga modeliramo**

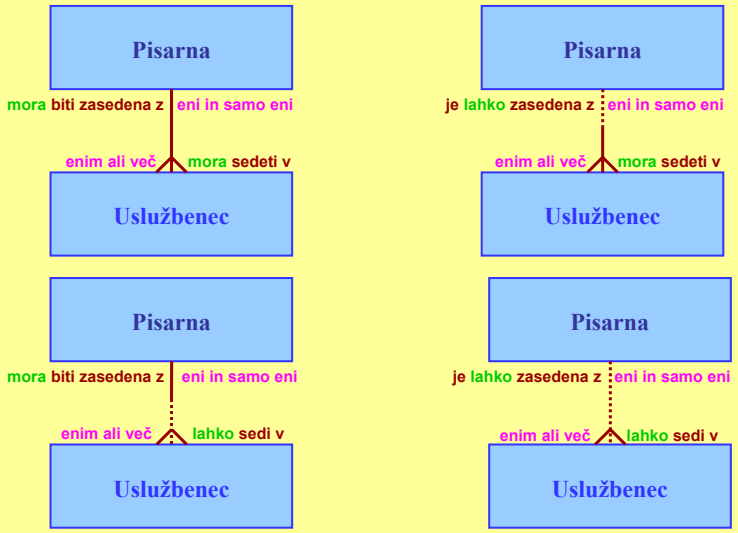
**Primer**: Naročilo in račun

- vsakemu naročilu ustreza svoj račun
- račun se izstavi enkrat mesečno za vsa naročila v tem časovnem obdobju

**Primer**: Pisarna in uslužbenec

## SSADM: Logični podatkovni model








## SSADM: Logični podatkovni model



**Posebnosti pri obravnavi razmerij**

**a) Razmerja M:N**

- pojavljajo se predvsem v zgodnjih fazah modeliranja
- kasneje jih preoblikujemo v razmerja 1:N, tako da vpeljemo nov entitetni tip

**Primeri:**

- Naročilo:Izdelek (nov entitetni tip Dobavnica)
- Pacient:Zdravilo (nov entitetni tip Recept)
- Učitelj:Predmet (nov entitetni tip Učitelj/Predmet)

**Atributi novega entitetnega tipa**

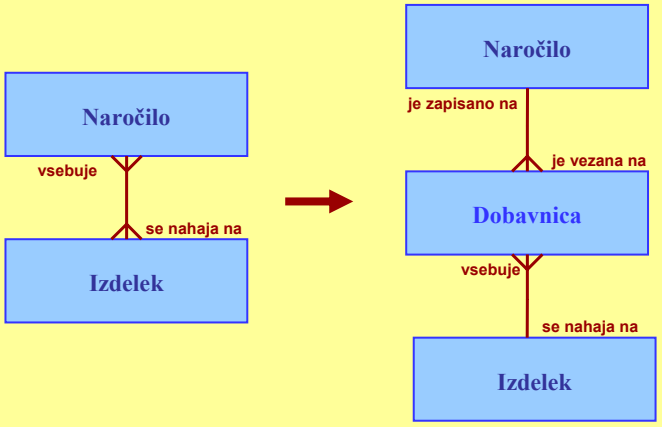
- primarna ključa obeh entitetnih tipov

**Poimenovanje novega entitetnega tipa**

- primerno ime ali [ime prve entitete] / [ime druge entitete]

## SSADM: Logični podatkovni model



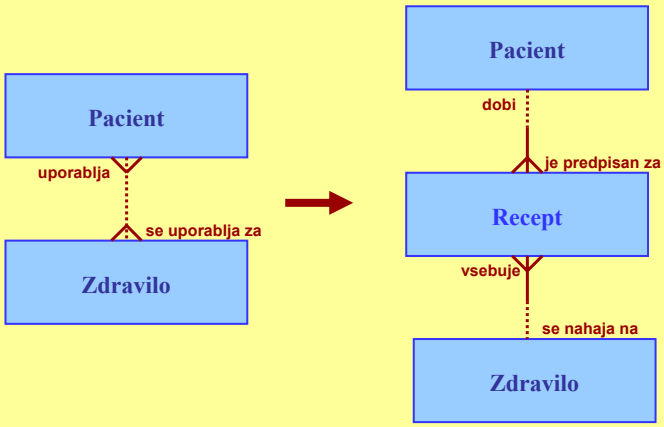






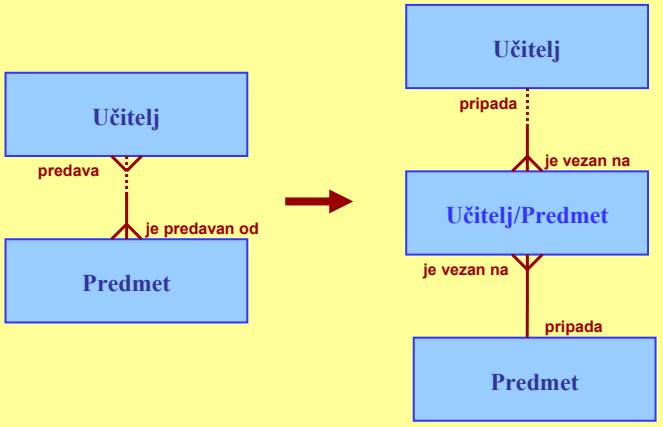
## SSADM: Logični podatkovni model





## SSADM: Logični podatkovni model








## SSADM: Logični podatkovni model



**b) Razmerja 1:1**

- ena entiteta je A povezana z natanko eno entiteto B
- združimo entitete, kjer lahko

**Primeri:**

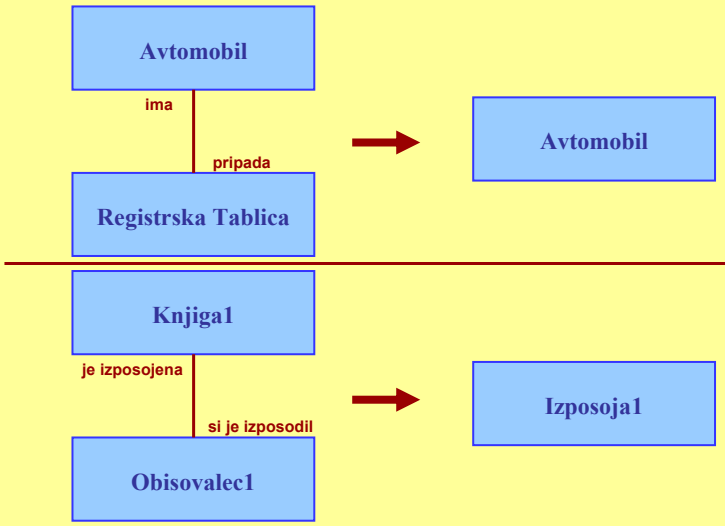
- Avtomobil : Registrska tablica
- Knjiga1 : Obiskovalec1 (samo en izvod vsake knjige, vse knjige izposojene, vsak obiskovalec ima izposojeno natanko eno knjigo)
- Knjiga2 : Obiskovalec2 (samo en izvod vsake knjige, ni nujno, da so izposojene vse knjige, vsak obiskovalec ima izposojeno natanko eno knjigo)
- Knjiga3 : Izposoja3 : Obiskovalec3 (samo en izvod vsake knjige, ni nujno, da so izposojene vse knjige, vsak obiskovalec ima lahko izposojenih več knjig)

**Atributi novega entitetnega tipa**

**Poimenovanje novega entitetnega tipa**

## SSADM: Logični podatkovni model



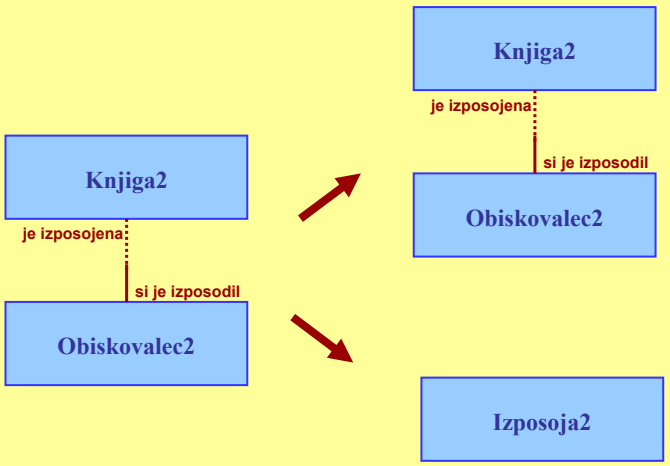






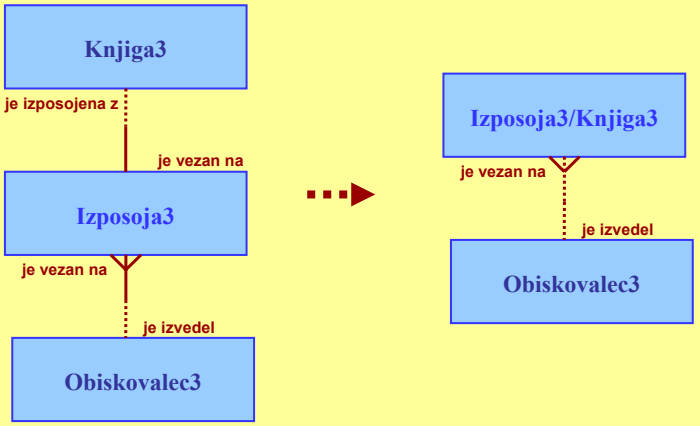
## SSADM: Logični podatkovni model





## SSADM: Logični podatkovni model







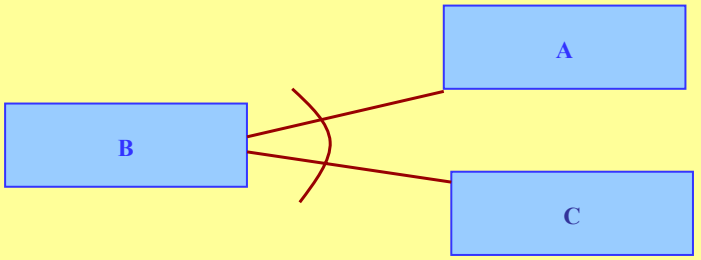


## SSADM: Logični podatkovni model



**c) Ekskluzivna razmerja:** eno razmerje izključuje drugo

**Če obstaja razmerje med A in B, ne more obstajati razmerje med C in B.**

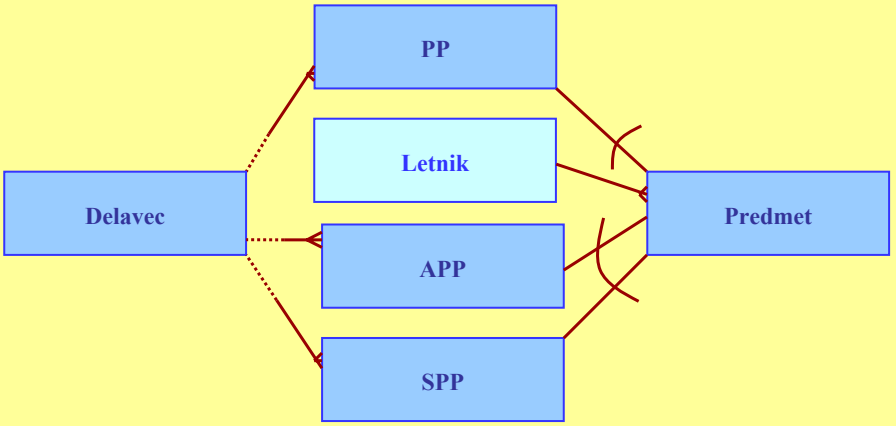



Primeri:

- Pacient, Soba, Čakalnica
- Predmet, Delavec, PP, APP, SPP

## SSADM: Logični podatkovni model









## SSADM: Logični podatkovni model

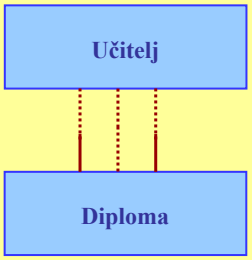


**d) Večkratna razmerja**

- med istim parom entitet obstaja več kot eno razmerje
- narišemo vsako razmerje posebej in ga poimenujemo

**Primeri:**

- Učitelj:Diploma (mentor, somentor, član komisije)
- Študent:Občina (stalnega, začasnega bivališča, zaposlitve)



## SSADM: Logični podatkovni model

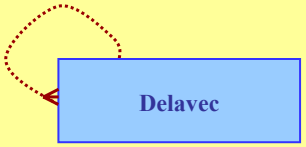


**e) Rekurzivno razmerje 1:N**

**Primer:** Delavec:Delavec, manager of, managed by

Relacijska shema:

Delavec (št.delavca, št.nadr.delavca, priimek, ime, naslov)







## SSADM: Logični podatkovni model



**f) Rekurzivno razmerje M:N**

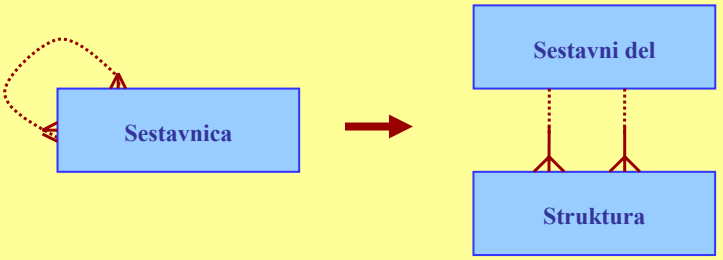
SSADM zahteva, da vpeljemo nov povezovalni entitetni tip (link entity)

**Primer:** Sestavnica (bill-of-materials)

Relacijska shema:

Sestavni del (šifra dela, ime dela, cena,...)

Struktura (šifra nadr.dela,šifra podr.dela, količina)



**g) Rekurzivno razmerje 1:1**

**Primer:** poročeni delavci znotraj iste organizacije

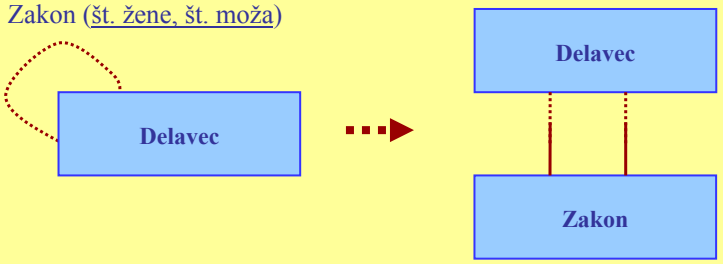
Relacijska shema:

Delavec (št.delavca, priimek, ime, naslov, št. zakonca)

ali

Delavec (št.delavca, priimek, ime, naslov)

Zakon (št. žene, št. moža)





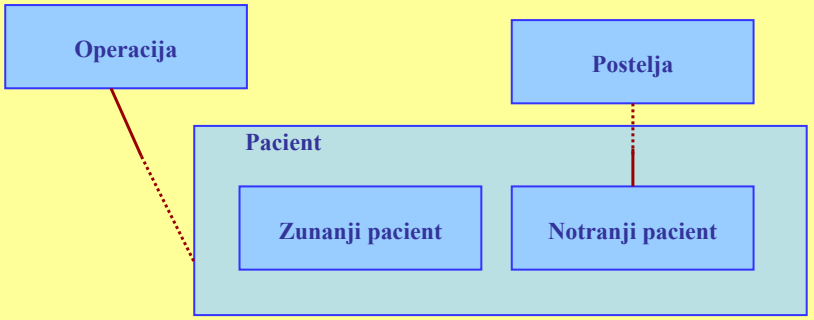



**h) Vsebovani entitetni tipi**

- za predstavitev več ravni podrobnosti

**Primer:** pacienti bolnišnice so dveh vrst:

- Notranji pacienti: tisti, ki po operaciji (ob)ležijo v bolnišnici
- Zunanji pacienti: tisti, ki po (manjši) operaciji odidejo domov



## SSADM: Logični podatkovni model



### Napotki za risanje diagramov, ki prikazujejo logično strukturo podatkov

- razmestitev entitet naj bo taka, da se razmerja ne križajo:
  - izogibanje križanju z lomljenimi črtami
  - uporaba samo pravokotnih črt ni priporočljiva
  - most, kadar se ni moč izogniti križanju
- entitete, ki spadajo v isto področje, naj bodo skupaj:
  - razmerja so največkrat med entitetami iz istega področja
  - projekt je lažje razdeliti na podprojekte
- vrane naj stojijo na nogah (nič mrtvih vran)
  - "master" entiteta zgoraj, "detail" entiteta spodaj
  - tega dogovora se ni moč vedno držati: včasih povzroči več križanja črt





## SSADM: Logični podatkovni model



### Validacija logične podatkovne strukture (LDS)

- LDS mora podpirati vse potrebne obdelave (zahteve po procesiranju podatkov), ki so izražene z DFD
- LDS služi kot osnova za navigacijo
- navigacija preko skupnih atributov
- dodatna dokumentacija (SSADM predlaga ustrezne obrazce) :
  - opis entitet
  - opis razmerij
  - opis atributov

**Alternativne notacije**

Chen: E-R diagrami

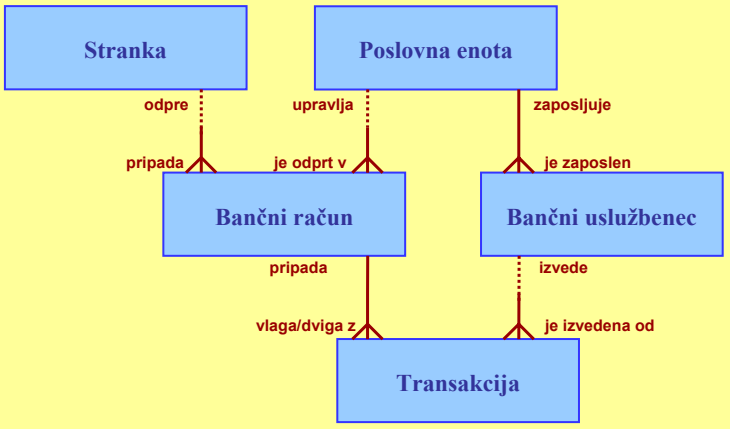
- entiteta
- atribut
- razmerje
- številnost

James Martin: Information Engineering

- entiteta
- razmerje
- opcijsko, obvezno







## SSADM: Model podatkovnih tokov



**Pojem model v SSADM:**

- del dokumentacije, ki odraža določen pogled na sistem
- običajno obsega ustrezne diagrame in spremljajočo tekstualno informacijo

**Model podatkovnih tokov sestavljajo:**

- množica diagramov podatkovnih tokov (DFD-Data Flow Diagrams)
- spremljajoča dokumentacija (kot dodaten opis, ki dopolnjuje diagrame)
  - opis zunanjih entitet
  - opis elementarnih procesov
  - opis vhodnih in izhodnih tokov



## SSADM: Model podatkovnih tokov



**Data Flow Diagram:** predstavitev informacijskih tokov v sistemu s pomočjo ustreznega diagrama

- kako informacije vstopajo in izstopajo iz sistema
- kako se informacije spreminjajo (procesi)
- kje se informacije v sistemu hranijo (data store)

Modeli podatkovnih tokov so pomembna tehnika v skoraj vseh strukturiranih metodah za analizo in načrtovanje.

**Pomembni so za:**

- določitev mej sistema (diagrami jasno odražajo meje in obseg sistema, ki ga predstavljajo)
- zagotavljanje popolnosti analize: izdelava diagramov in navzkrižna primerjava z ostalimi tehnikami SSADM zagotavlja, da so upoštevani vsi informacijski tokovi, podatkovne shrambe in aktivnosti v sistemu

## SSADM: Model podatkovnih tokov



**Pomembni so za (nadaljevanje):**

- osnova za načrt sistema:
  - DFD prikazujejo glavna funkcijska področja;
  - omogočajo razpravo o različnih rešitvah z različno funkcionalnostjo;
  - na podlagi dogovorjenega načrta na najvišjem nivoju jih uporabljamo za podrobnejše načrtovanje
- identifikacija funkcij:
  - v načrtovanju,
  - v okviru faze Specifikacija zahtev,
  - omogočajo identifikacijo funkcij, ki služijo kot osnova za načrtovanje procesiranja (programov)



## SSADM: Model podatkovnih tokov



**Pomembni so za (nadaljevanje):**

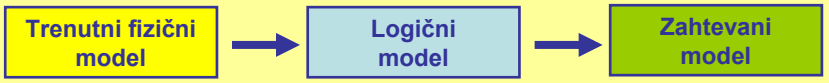
- komunikacija z uporabniki sistema:
  - predstavljajo netehnični opis, ki je razumljiv uporabniku;
  - služijo kot osnova za diskusijo;
  - služijo za verifikacijo ...
- opis sistema na več nivojih:
  - **najvišji nivo:** celoten sistem brez podrobnosti, pomemben predvsem za določitev mej sistema in povezav z drugimi sistemi ter uporabniki
  - **nižji nivoji:** podroben opis posameznih procesov
- delitev dela pri velikih projektih: na podlagi skupnega diagrama na najvišjem nivoju poteka paralelno razvoj diagramov na nižjih nivojih

## SSADM: Model podatkovnih tokov



**Tri vrste modelov podatkovnih tokov v SSADM**

- **trenutni fizični model** (Current Physical): model obstoječega sistema
- **logični model** (Logical): bistvene aktivnosti obstoječega sistema se izločijo (izluščijo) iz fizičnega modela
- **zahtevani model** (Required): model bodočega sistema




Na podlagi logičnega modela in seznama zahtev se izdela načrt sistema na najvišjem nivoju (Selected Business System Option), ki se naprej razvije v množico DFD, ki predstavljajo nov sistem.



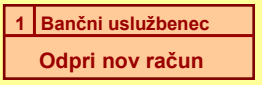
## SSADM: Model podatkovnih tokov



**Komponente diagramov podatkovnih tokov**

a) zunanje entitete (external entities)


b) procesi (processes)




c) podatkovne shrambe (data stores)


d) podatkovni tokovi (data flow)

## SSADM: Model podatkovnih tokov



**a) Zunanje entitete:**

- vsak izvor ali ponor informacij iz sistema (kdor daje informacije sistemu in kdor jih od sistema prejema)
- vsa informacija, ki je predstavljena v sistemu, mora na začetku priti od neke zunanje entitete

**Primeri:** uporabnik sistema, zunanja organizacija, drug računalniški sistem

- simbol: elipsa
- oznaka (identifikator): mala tiskana črka




## SSADM: Model podatkovnih tokov

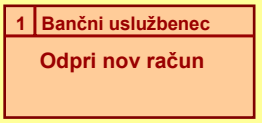


**b) Proces:**

- transformira in izvaja manipulacije nad podatki v sistemu
- simbol: pravokotnik, ki vsebuje
  - ime procesa
  - oznako (identifikator) procesa: zaporedna številka
  - v fizičnih DFD tudi lokacijo ali izvajalca (osebje, zadolženo za izvedbo procesa)

**Ime procesa je velelni stavek:**

- pravilno: zabeleži novega kupca
- narobe: nov kupec, registracija



## SSADM: Model podatkovnih tokov



**c) Podatkovna shramba**

- označuje mesto, kjer se informacija v sistemu shrani (for a time)
- simbol: škatla brez desne stranice

Različne vrste shramb v različnih modelih

- v trenutnem fizičnem modelu so prikazane dejanske shrambe podatkov, npr. kartice, računalniške datoteke ipd.
- v logičnem in zahtevanem modelu so shrambe računalniške
- v vseh primerih so lahko shrambe začasne/prehodne (angl. transient)

**Oznake shramb**

- M: ročna shramba
- D: računalniška shramba
- T: začasna shramba
- T(M): začasna ročna shramba





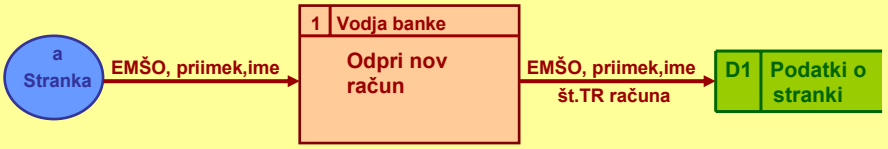
## SSADM: Model podatkovnih tokov



**d) Podatkovni tok**

- predstavlja paket podatkov, ki teče med objekti na DFD
- simbol: usmerjena puščica, ki označuje smer toka podatkov
- oznaka: ime ali kratek opis podatkov, ki nastopajo v toku

**Primer:**



## SSADM: Model podatkovnih tokov



**Pomembne karakteristike podatkovnih tokov**

- Na enem koncu podatkovnega toka mora biti vedno proces (podatkovni tok teče vedno ali v proces ali iz procesa)
- Na drugem koncu je lahko
  - zunanja entiteta
  - podatkovna shramba
  - nek drug proces

**Včasih imamo opravka s podatkovnimi tokovi med zunanjimi entitetami:**

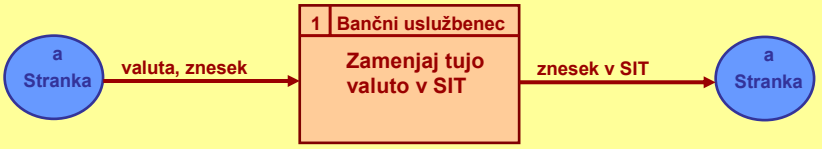
- čeprav so ti tokovi zunaj sistema, jih vseeno prikažemo zaradi večje razumljivosti
- namesto neprekinjene uporabimo črtkano črto 




## SSADM: Model podatkovnih tokov



- V fizičnem modelu podatkovni tokovi predstavljajo dejanske tokove, npr. obrazce, ki potujejo med oddelki, ali telefonske razgovore med nekom znotraj sistema in nekom zunaj sistema
- V logičnem in zahtevanem modelu tokovi predstavljajo:
  - atribute, ki jih potrebuje nek proces ali
  - izhod iz procesa (rezultat)



## SSADM: Model podatkovnih tokov



**Konstruiranje (risanje) DFD**

**Primer**: Zahtevani DFD za najvišji nivo enostavnega bančnega sistema

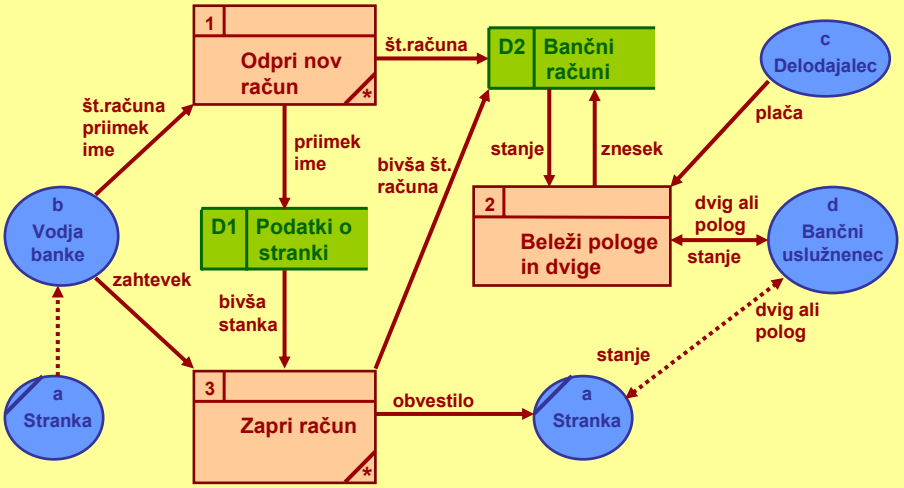
**Sistem obsega:**

- odpiranje novih računov (registracija novih strank):
  - stranka odda zahtevek vodji banke
  - vodja izvede vnos podatkov in odpre račun
- beleženje dvigov in pologov:
  - postopek se izvede preko bančnega uslužbenca
  - polaga stranka sama ali delodajalec (plačo), dviga samo stranka
- zapiranje računov
  - zahtevek odda stranka vodji banke
  - ob zaprtju računa obvestilo stranki




## SSADM: Model podatkovnih tokov





## SSADM: Model podatkovnih tokov



**Razlika med prikazovanjem zunanjih entitet v fizičnem in logičnem/zahtevanem modelu**:

- Na začetku meje sistema niso še natančno definirane.
- V fizičnem modelu gremo raje nekoliko v širino, zato so običajno izvajalci operacij sestavni del sistema in so prikazani v simbolu za proces.
- V logičnem/zahtevanem modelu se meja izkristalizira: dejansko podatke o kupcu vnaša vodja banke, ki je sedaj zunanja entiteta, tok podatkov med stranko in njim je prikazan črtkano.
- Če bi podatke o stranki vnašala stranka sama, bi bil seveda diagram drugačen.
- Podobno velja za interakcijo med stranko in bančnim uslužbencem pri pologih in dvigih.




## SSADM: Model podatkovnih tokov



**Številčenje procesov:** ne vpliva na vrstni red izvajanja

Podvajanje podatkovnih shramb in zunanjih entitet: zaradi večje preglednosti (da se izognemo križanju podatkovnih tokov)

- podvojene podatkovne shrambe: dvojna črta na levi
- podvojene zunanje entitete: poševna črta na levem zgornjem delu

**Izgled diagrama:**



- zunanje entitete ob robu
- podatkovne shrambe v sredini
- max. 12 procesov na enem diagramu (boljše pravilo: cca 7)

DFD je namenjen komunikaciji z uporabnikom, zato sta jasnost in čitljivost enako pomembni kot tehnična vsebina.

## SSADM: Model podatkovnih tokov



### Nivoji DFD diagramov

Vsak proces lahko razbijemo na več enostavnejših procesov, ki jih opišemo z DFD na naslednjem nivoju.

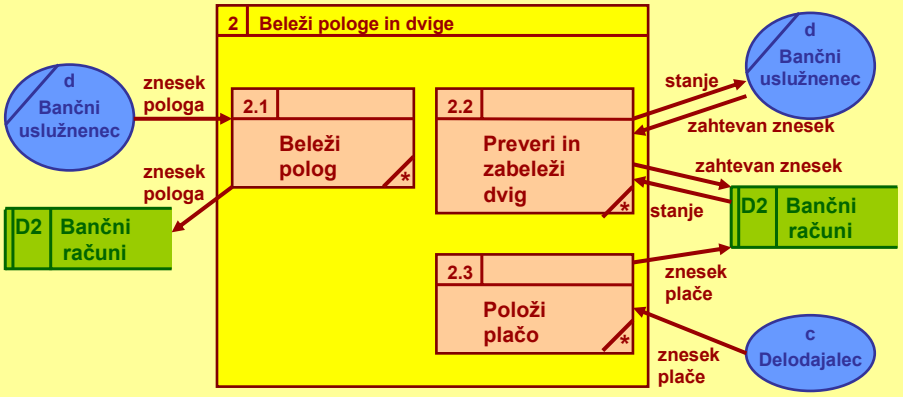
### Nivoji DFD:

- nivo 1 top-level diagram, ki opisuje celoten sistem
- nivo 2 podrobnejši opis procesov z nivoja 1
- nivo 3 podrobnejši opis procesov z nivoja 2
- itd.




## SSADM: Model podatkovnih tokov





## SSADM: Model podatkovnih tokov



**Do katerega nivoja je treba izvršiti dekompozicijo:**

- odvisno od kompleksnosti procesa
- za enostavne procese sploh ni potrebna
- kompleksne procese razbijamo toliko časa, dokler ne pridemo do dovolj enostavnih
- procesom, ki niso več razbiti naprej, pravimo elementarni procesi
- na elementarne procese lahko naletimo na vseh nivojih
- elementarni procesi so na DFD označeni z zvezdico
- pri dekompoziciji ne pride samo do razbitja procesov ampak tudi do razbitja podatkovnih tokov (prej enoten tok iz zunanje entitete lahko razpade na več ločenih, bolj natančno opredeljenih podatkovnih tokov)



## SSADM: Model podatkovnih tokov



**Označevanje procesov na nižjih nivojih:**

| | |
|---|---|
| • nivo 1 | 3 |
| • nivo 2 | 3.1 3.2 3.3 |
| • nivo 3 | 3.3.1 3.3.2 3.3.3 3.3.4 |

**Primer:** dekompozicija procesa št. 2 iz prejšnjega DFD

- okvir označuje proces z višjega nivoja; številka procesa in ime sta na vrhu okvirja
- razbitje podatkovnih tokov: tokovi z višjega nivoja se ali podvojijo ali razpadejo na več tokov
- pri dekompoziciji je treba dvosmerne tokove nadomestiti z dvema ločenima tokovoma (dvosmerni tokovi so dovoljeni samo na nivoju 1)

## SSADM: Model podatkovnih tokov



**Dodatna dokumentacija, ki spremlja DFD:**

- opis elementarnih procesov
- opis zunanjih entitet
- opis vhodov in izhodov (I/O Descriptions):

  podrobnejši opis podatkovnih tokov na najnižjem nivoju, ki prečkajo meje sistema (kateri atributi nastopajo)





## SSADM: Diagram sprememb stanja entitet



**Diagram stanja entitet = Entity Life History diagram (ELH)**

**Splošno o diagramih sprememb stanja entitet**

ELH modelirajo sistem s stališča, kako se informacije v njem spreminjajo:

- prikazujejo vse možne spremembe informacij, ki se lahko zgodijo v sistemu
- skupaj s kontekstom vsake spremembe

**ELH narišemo za vsako entiteto posebej:**

- je diagramski prikaz življenja entitete od njenega kreiranja do brisanja
- življenje entitete je prikazano kot zaporedje (dovoljenih) dogodkov, ki povzročajo spremembe entitete
- dogodek je karkoli, kar sproži proces, ki povzroči spremembo entitete
- pri ELH je poudarek na modeliranju dogodkov, ne procesov

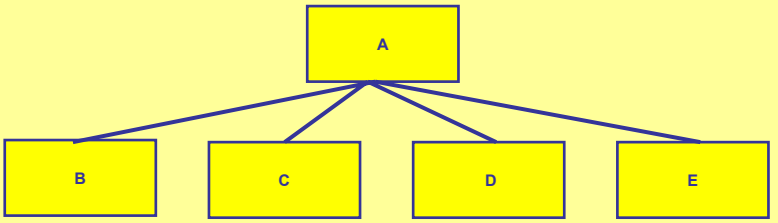
## SSADM: Diagram sprememb stanja entitet



**Za risanje ELH diagramov se uporabljajo ti. Jacksonove strukture oziroma Jacksonovi diagrami:**

- zaporedje ali sekvenca

  A je zaporedje komponent B, C, D in E.



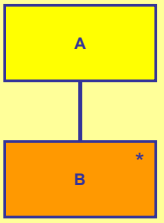



## SSADM: Diagram sprememb stanja entitet



- ponavljanje ali iteracija

  A je ponavljanje komponente B.

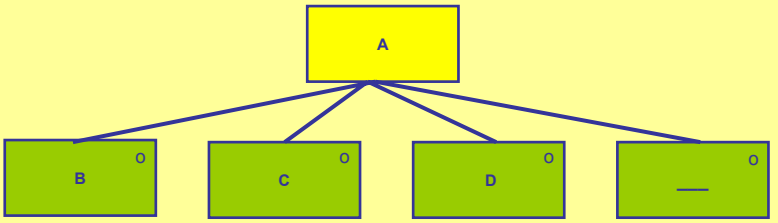


## SSADM: Diagram sprememb stanja entitet



- izbira ali selekcija

  A je izbira med komponentami B, C in D, lahko pa ne izberemo tudi nobene izmed treh – ničelna izbira (null box)





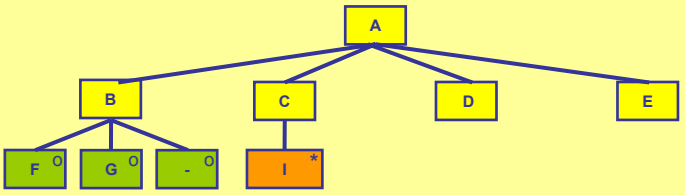

## SSADM: Diagram sprememb stanja entitet



Jacksonove diagrame beremo od zgoraj navzdol in od leve proti desni

- običajno jih gradimo postopoma
- po potrebi vključimo tudi ničelno izbiro (null box)

**Možna zaporedja v izvajanju komponent:**



## SSADM: Diagram sprememb stanja entitet



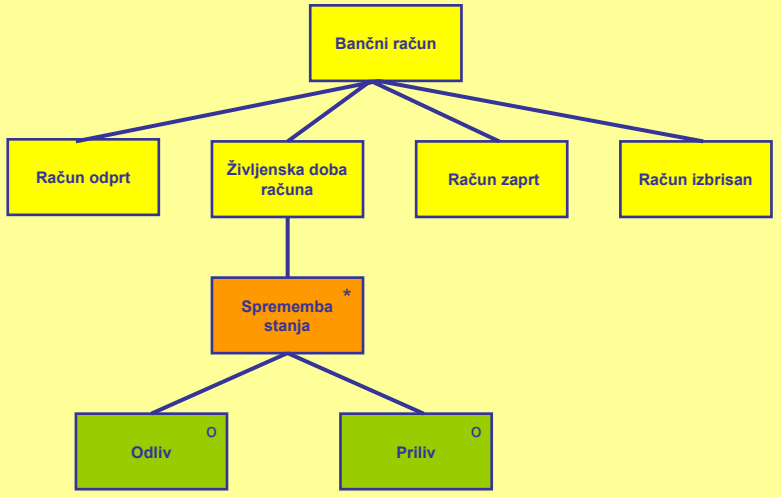
**Primer: ELH za Bančni račun**

- RAČUN ODPRT:
  prvi dogodek v življenju (dejanske) entitete je vedno kreiranje
- ŽIVLJENSKA DOBA RAČUNA:
  sledi zaporedje transakcij (polog, dvig, vnovčitev čeka, nakazilo OD, plačilo trajnih nalogov):
  - SPREMENI STANJE: vsaka izvedena transakcija
  - PRILIV ali ODLIV: transakcija, polog ali dvig denarja na račun
- RAČUN ZAPRT:
  življenje se konča z zaprtjem računa
- RAČUN IZBRISAN:
  brisanje poskrbi za izbris vseh podatkov o računu

## SSADM: Diagram sprememb stanja entitet





## SSADM: Diagram sprememb stanja entitet



**Uporaba Jacksonovih diagramov v SSADM: za več namenov**

- v ELH za modeliranje vrstnega reda dogodkov, ki vplivajo na podatke v sistemu
- v I/O Structure Diagrams (diagrami vhodno izhodnih struktur) za modeliranje položaja podatkov v vhodih in izhodih iz sistema
- v Update in Enquiry Process Models za prikaz zaporedja podprogramov, ki jih izvrši računalnik med izvajanjem določenega procesa

**Pristop k risanju ELH diagramov**

Matrika Dogodek/Entiteta

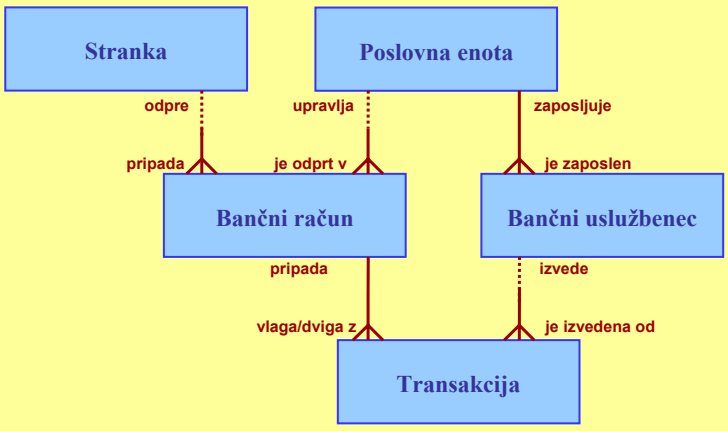
- vsak dogodek lahko učinkuje na več entitet
- na vsako entiteto lahko učinkuje več dogodkov



## SSADM: Diagram sprememb stanja entitet



**Podatkovni model:**



## SSADM: Diagram sprememb stanja entitet



| ENTITETA / DOGODEK | Stranka | Bančni račun | Transakcija |
|---|---|---|---|
| Račun odprt | C | C | |
| Priliv | | M | C |
| Odliv | | M | C |
| Račun zaprt | | M | |
| Račun izbrisan | | D | |

**C** **kreiraj (Create)**
**M** **spremeni (Modify)**
**D** **izbriši (Delete)**



## SSADM: Diagram sprememb stanja entitet



**Poimenovanje vozlišč v diagramu ELH**

- učinek (angl. effect): izraz, s katerim opišemo interakcijo med dogodkom in entiteto
- listi ustrezajo učinkom posameznih dogodkov (effect boxes)
- čeprav prikazujejo učinek, je ime lista enako imenu dogodka
- notranja vozlišča (nodes) izražajo veljavna zaporedja dogodkov (v smislu Jacksonovih diagramov grupirajo več podrejenih vozlišč v zaporedje, izbiro ali ponavljanje)
- imena notranjih vozlišč odražajo določeno stanje v življenju entitete
- koren predstavlja diagram kot celoto
- ime korena je enako imenu entitete

## SSADM: Diagram sprememb stanja entitet



**Posebni primeri, ki lahko nastopajo v diagramih ELH**

- paralelene strukture
- izhodi (quit and resume)
- izhodi zaradi naključnih dogodkov
- brezpogojni izhodi
- dogodki z različnimi učinki (effect qualifiers, optional effects)
- sočasni učinki (entity roles, simultaneous effects)

**Paralelne strukture**

- kadar vrstnega reda posameznih dogodkov ali notranjih vozlišč ni moč natančno napovedati
- POZOR: ne gre za sočasni učinek dveh dogodkov

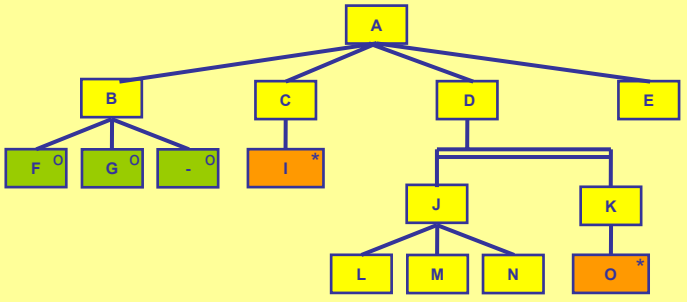



## SSADM: Diagram sprememb stanja entitet

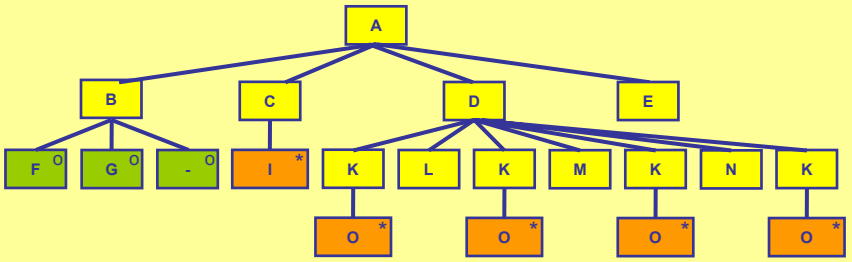


**Primer:**

- med zaporedjem dogodkov L, M, N se poljubno mnogokrat pojavi dogodek O
- komponenta D je paralelna struktura



## Odprava paralelne strukture



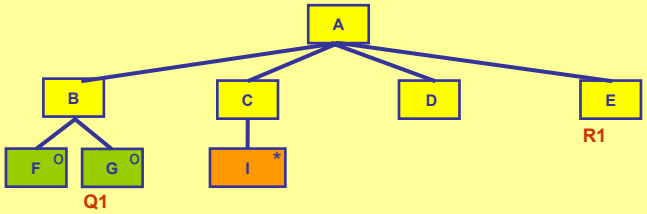





## Izhodi (quit and resume)

- uporaba: da se izognemo kompleksnim diagramom
- omogočajo izstop iz enega dela diagrama in nadaljevanje na drugem delu diagrama ELH
- izhod se označi s Q (Quit) in številko na desni strani pravokotnika
- nadaljevanje se označi z R (Resume) in številko na levi strani pravokotnika

**Primer:** ponovno odprtje računa



## SSADM: Diagram sprememb stanja entitet



**Izhodi zaradi naključnih dogodkov**

- naključni dogodek se lahko pojavi kadarkoli v življenju entitete in povzroči nadaljevanje na točno določenem mestu
- mesta izhoda (tj. Q) ne moremo označiti; namesto tega na dnu diagrama samo opišemo ta dogodek
- mesto nadaljevanja označimo enako kot prej
- nadaljevanje je lahko opisano kot posebna samostojna struktura

**Primer:** smrt imetnika računa



## SSADM: Diagram sprememb stanja entitet




Q2: izhod od kjerkoli ob smrti stranke, nadaljevanje R2

## SSADM: Diagram sprememb stanja entitet

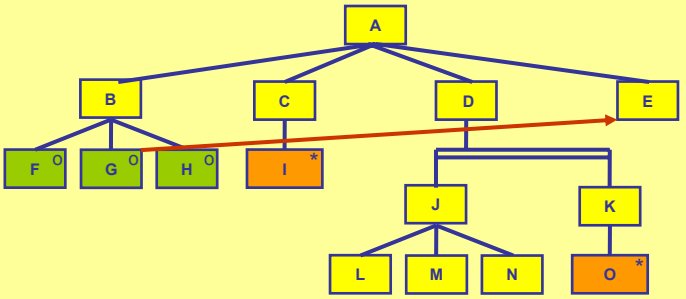


**Brezpogojni izhod:**

- ni dovoljen

**Primer napačnega izhoda:** po izvršitvi G naj se izvrši E

**Rešitev:** drugačen diagram

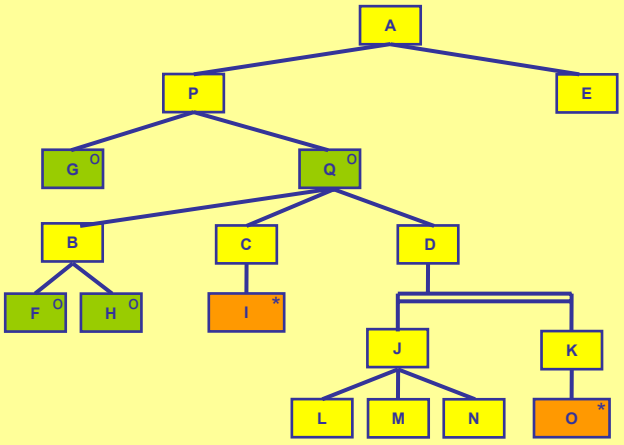





## SSADM: Diagram sprememb stanja entitet



**Rešitev:** spremenimo diagram



## SSADM: Diagram sprememb stanja entitet



**Dogodki z različnimi učinki:**

- effect qualifiers, optional effects
- dogodek lahko učinkuje na entiteto na več različnih načinov

**Primer:** dogodek Odliv (Debit) učinkuje na entiteto Bančni račun na 2 načina:

- če je na računu dovolj sredstev, se dvig izvede
- če na računu ni dovolj sredstev, se dvig izvede samo v primeru, če ni presežen dovoljeni limit (določena vrednost, za koliko se lahko stranka zadolži pri banki)

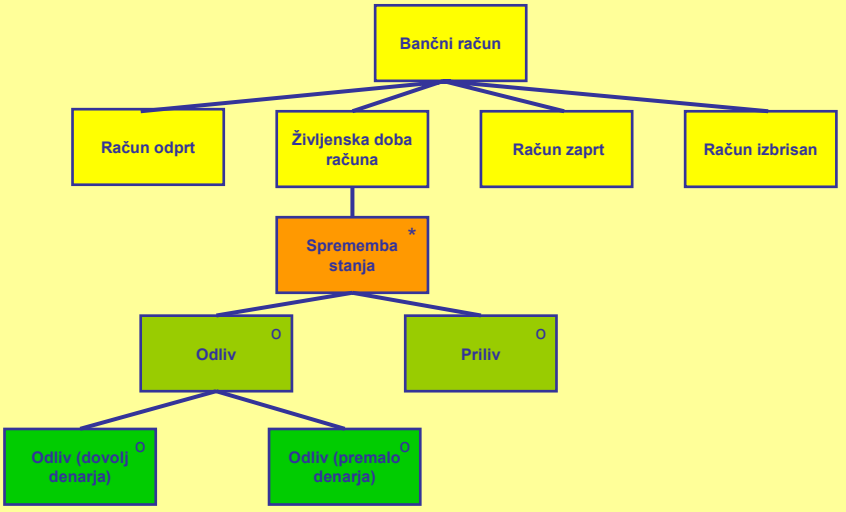
**V diagramu ELH:**

- narišemo možne učinke kot selekcijo
- imenu dogodka dodamo opis (opredelitev, qualifier) v okroglem oklepaju




## SSADM: Diagram sprememb stanja entitet





## SSADM: Diagram sprememb stanja entitet



**Sočasni učinki:**

- entity roles, simultaneous effects
- dogodek lahko učinkuje na več konkretnih entitet istega tipa

**Primer:** ob zaprtju bančnega računa lahko:

- denar dvignemo
- prenesemo sredstva na nov račun

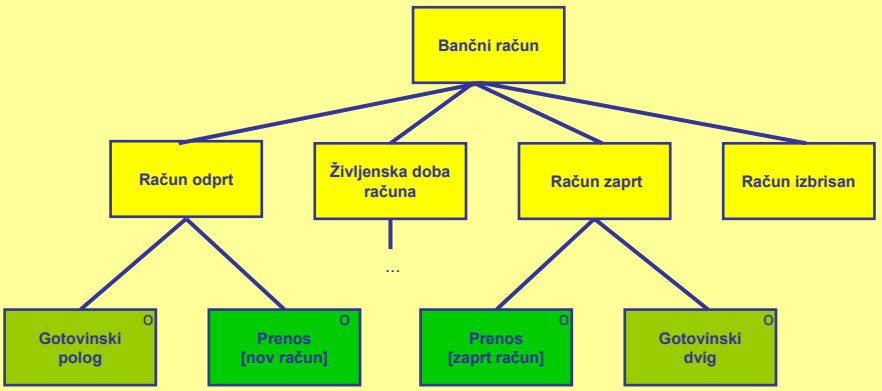
**V diagramu ELH:**

- dogodek se pojavi dvakrat v diagramu ELH, zato je potrebno pojasnilo v oglatem oklepaju
- v oglatem oklepaju napišemo, na katero konkretno entiteto učinkuje dogodek (entity role)



## SSADM: Diagram sprememb stanja entitet





## SSADM: Diagram sprememb stanja entitet



### Operacije

- operacije podrobno opisujejo, kaj se zgodi ob vsakem dogodku
- vsaka operacija spremeni vrednost vsaj enega atributa in/ali spremeni razmerje
- operacije služijo kot osnova za podrobno specifikacijo procesov (prenesejo se v Update Process Models)
- tipično so operacije splošno definirane in brez podrobnosti
- s pomočjo operacij lahko razberemo zaporedje stavkov programa, ki izvaja aktivnosti nad entiteto

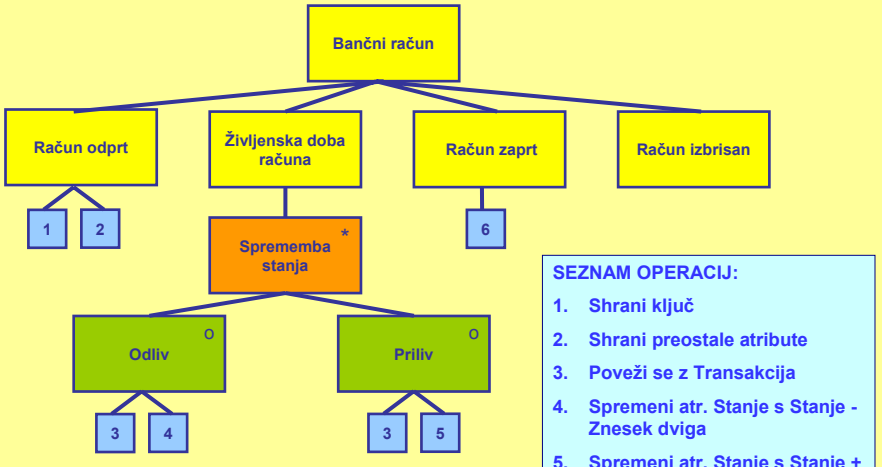
### Primer:

- aktivnosti nad entiteto Bančni račun




## SSADM: Diagram sprememb stanja entitet

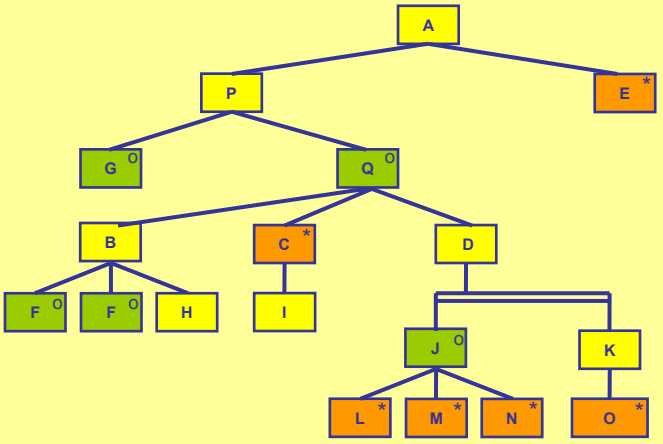




## SSADM: Diagram sprememb stanja entitet



### Primer napačne strukture ELH diagrama

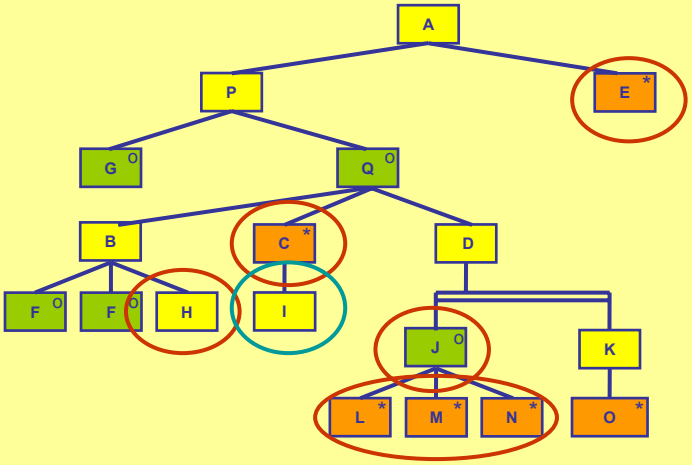





## SSADM: Diagram sprememb stanja entitet



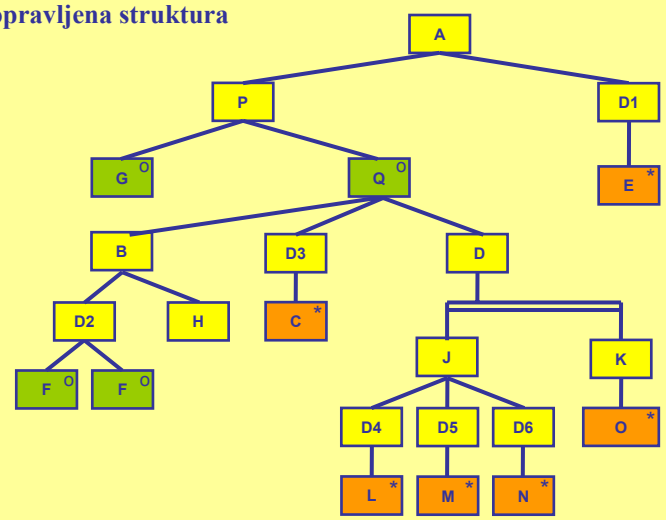
### Primer napačne strukture ELH diagrama – z oznakami



## SSADM: Diagram sprememb stanja entitet



**Popravljena struktura**






## SSADM: Povezave med tremi pogledi na sistem



**Princip SSADM:**

**Vsi trije pogledi na sistem so tesno povezani med sabo, ker izhajajo iz istega izhodišča in se nanašajo na isti končni rezultat**

Ključnega pomena so stične točke, recimo:

- entitete iz LDM in podatkovne shrambe iz DFD
- dogodki iz ELH in procesi iz DFD
- entitete iz LDM in stanja entitet iz ELH

Vsak vidik prikazuje tudi del zgodbe, ki jo drugi vidiki ne zmorejo prikazati

**Dve načeli: popoln opis** in **izločanje nasprotuj iz opisov**

**Stopnja povezanosti je odvisna od faze dela:**

- nanaša se predvsem na specifikacije novega sistema
- v obstoječem sistemu so možne anomalije

## SSADM: Povezave med tremi pogledi na sistem



**Zveza med DFM in LDM modelom**

**Povezava med entitetami in podatkovnimi shrambami:**

- vsaka podatkovna shramba je povezana z eno ali več entitetami
- neka entiteta se lahko pojavi samo v eni podatkovni shrambi
- formalna dokumentacija: Logical Data Store/Entity Cross-Reference Form

**Izjeme**:

- v obstoječem sistemu se podatki pogosto podvajajo, zato lahko ista entiteta nastopa v več podatkovnih shrambah
- prehodne podatkovne shrambe ne predstavljajo trajno shranjenih podatkov, zato niso povezana z entitetami v LDS





## SSADM: Povezave med tremi pogledi na sistem



**Zveza med DFM in LDM modelom (nadaljevanje)**

**Povezava med entitetami in atributi v podatkovnih tokovih:**

- Atributi v podatkovnih tokovih morajo pripadati entitetam v LDS
- formalna kontrola: navzkrižna primerjava DFD in Entity Descriptions

**Izjeme**:

- prehodni elementi kot npr. delni rezultati
- izračunane vrednosti, ki se izpisujejo
- sporočila o napakah (če se nek podatkovni tok nanaša samo na sporočilo o napaki)

## SSADM: Povezave med tremi pogledi na sistem



### Zveza med LDM modelom in ELH

Pri izdelavi ELH in preverjanju popolnosti si pomagamo z LDS in spremljajočim opisom entitet (Entity Description), ki vsebuje seznam atributov:

- za vsako entiteto svoj diagram ELH
- dogodki (in z njimi povezane operacije) v ELH diagramu morajo kreirati, kasneje dodati in brisati vse atribute
- spremembe razmerij (npr. sprememba master entitete) so lahko posledica dogodkov, ki jih je treba upoštevati (pripomoček pri identifikaciji dogodkov)
- na ELH diagrame entitet lahko vplivajo ELH diagrami entitet, ki so z njimi v razmerju kot "master" ali "detail" (npr. brisanje računa ima lahko za posledico tudi brisanje transakcij)





## SSADM: Povezave med tremi pogledi na sistem



### Zveza med DFM in ELH

Obstaja povezava med dogodki v ELH in procesi v DFD:

- dogodki sprožijo procese, ki spreminjajo vrednosti podatkov v sistemu
- vsak proces na najnižjem nivoju DFD, ki spremeni vrednost podatkov v podatkovni shrambi, se sproži s pomočjo nekega dogodka, ki je prikazan v ELH (z vsakim procesom ažuriranja je povezan dogodek)
  - če je izvor dogodka izven sistema, je največkrat povezan z vhodnim tokom podatkov.
  - če je proces sprožen na drugačen način (ne z vhodnim tokom podatkov), dogodek ni eksplicitno prikazan na DFD, ampak ga identificiramo s sklepanjem ob pregledu diagrama.

## SSADM: Povezave med tremi pogledi na sistem



**V praksi**:

- Najprej pregledamo DFD in skušamo ugotoviti dogodke, ki sprožijo posamezne procese.
- Ko narišemo ELH diagrame, skušamo na novo odkrite dogodke povezati s procesi (vpeljava novih funkcij).



## SSADM: Analiza zahtev



**ANALIZA ZAHTEV (REQUIREMENTS ANALYSIS)**

**1. Splošno**

Kaj naj bi nov sistem delal in kako dobro

- osnova: razumevanje obstoječega stanja
- poudarek: na bodočem sistemu (obstoječi sistem analiziramo do take globine, da lahko definiramo zahteve za nov sistem)

**2 fazi:**

- Analiza obstoječega stanja (Investigation of Current Environment)
- Poslovne opcije sistema (Business System Options)

**Rezultat**: več opcij, ki jih predstavimo najvišjemu vodstvu

## SSADM: Analiza zahtev



**Cilji analize zahtev:**

- osnovni: določiti zahteve za nov sistem
- določitev mej bodočega računalniško podprtega sistema
- jasno razumevanje, kako bo bodoči sistem zadovoljil poslovne potrebe
- jasno razumevanje stroškov in koristi
- jasna odločitev za nadaljevanje dela (izbor ustrezne opcije)
- vključevanje uporabnika: uporabnik je "lastnik" (soustvarjalec) sistema in je odgovoren za vse odločitve glede usmeritve projekta



## SSADM: Analiza zahtev



**Izdelki v fazi analize zahtev:**

1. Opis trenutnega stanja (Current Services Description)
2. Seznam uporabnikov (User Catalogue)
3. Seznam zahtev (Requirements Catalogue)
4. Izbrana poslovna opcija sistema (Selected Business Option)

**1. Opis trenutnega stanja (Current Services Description):**

- prikaz logične organizacije podatkov in postopkov v obstoječem sistemu
  - logični podatkovni model
  - logični model podatkovnih tokov

## SSADM: Analiza zahtev



**2. Seznam uporabnikov (User Catalogue):**

- opis del in nalog, ki jih bodo opravljali neposredni (on-line) uporabniki bodočega sistema

**3. Seznam zahtev (Requirements Catalogue):**

- opis in prioriteta zahtev, za katere smo se dogovorili z uporabnikom
- zajema tako funkcionalne kot nefunkcionalne zahteve
- za vsako zahtevo poseben obrazec

**4. Izbrana poslovna opcija (Selected Business Option):**

- prikaz optimalne rešitve postavljenih zahtev
- analiza stroškov in koristi

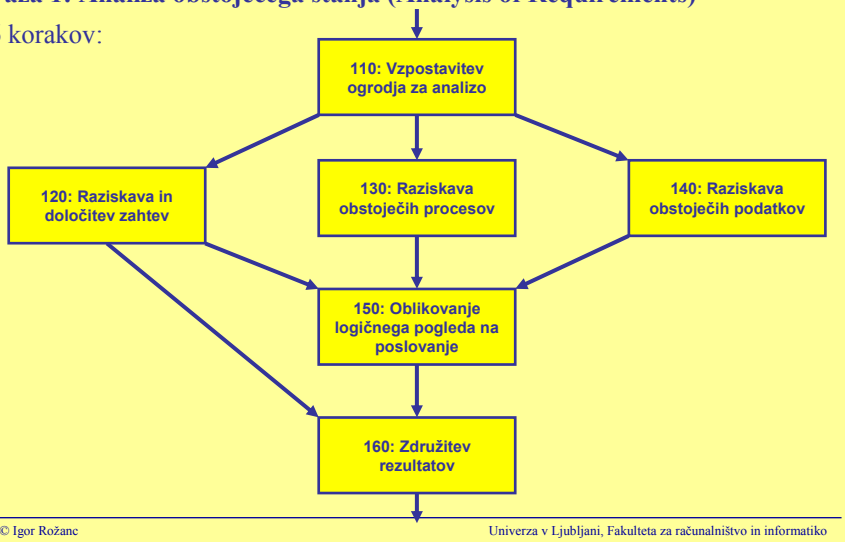


## SSADM: Analiza obstoječega stanja



**Faza 1: Analiza obstoječega stanja (Analysis of Requirements)**

6 korakov:



## SSADM: Analiza obstoječega stanja



**Uporabljene tehnike**

- intervjuji
- študij dokumentacije o obstoječem sistemu
- anketiranje
- opazovanje sistema v praksi
- uporaba rezultatov iz predhodnih študij

**Predstavitev rezultatov:**

- množica DFD, ki opisujejo obstoječi sistem
- LDM obstoječega sistema
- seznam zahtev
- dodatne informacije: obseg informacij (število), stroški (pridejo prav pri oceni stroškov in velikosti novega sistema)



## SSADM: Analiza obstoječega stanja



**Korak 110:**

**Vzpostavitev "ogrodja" za analizo (Establish analysis framework)**

Velik del tega koraka je posvečen vodstvenim (managerskim) zadevam:

- vzpostavitev trdnih temeljev za projekt
- določitev obsega in namena projekta
- izdelava plana za prve 3 faze

**2 ločena, a med seboj povezana dela:**

- vzpostavitev projekta
- pričetek analize

**Vhodi in izhodi za korak 110:** 2 kategoriji

- izdelki, povezani z vodenjem projekta
- zgodnji rezultati analize

## SSADM: Analiza obstoječega stanja



**V1 - Project Initiation Document (Dokument o vzpostavitvi projekta):**

- osnova za pričetek dela
- dogovor med predstavniki uporabnikov (naročnikom) in razvijalci programske opreme (izvajalci)
- opis osnovnih zahtev in pogojev za izvedbo
- v primeru predhodne študije izvedljivosti so lahko pogoji (sredstva, roki, meje sistema) natančno opredeljeni
- v nasprotnem primeru: grob opis obsega, omejitev, predvidenih stroškov in rokov za izvedbo

**V2 - Predhodne študije**

- naloge v tej fazi so odvisne od predhodnih študij
- če so bile narejene predhodne študije, se obseg dela zmanjša: izkoristimo lahko rezultate teh študij (seznam zahtev, začetni DFD in LDS diagrami)
- preveriti je treba, ali so rezultati predhodnih študij še veljavni



## SSADM: Analiza obstoječega stanja



**I1 - Mrežni diagram in opis aktivnosti, struktura in opis izdelkov, diagram toka izdelkov in namen projekta:** izdelki, povezani z vodenjem projekta po metodi PRINCE (PRojects IN a Controlled Environment): ne obravnavamo

**I2 - Pregledni (začetni) diagram podatkovnih tokov, začetni diagram logičnih podatkovnih struktur:** povezana z določitvijo mej sistema, ki ga bomo proučevali

- ne zajeti preveč
- ne izpustiti kakega pomembnega področja
- v večjih sistemih:
- z integracijo delnih diagramov: težko najti enega uporabnika, ki bi poznal celoto
- na sestanku s celotno skupino uporabnikov

## SSADM: Analiza obstoječega stanja



**I3 - Kontekstni diagram:**

- celoten sistem je prikazan kot en sam proces
- spada na nivo 0
- pomaga pri določitvi mej sistema (jasno so vidne zunanje entitete)

**I4 – Začetni seznam zahtev**

- "nastavitev" (inicializacija) seznama zahtev
- seznam zahtev je eden glavnih SSADM izdelkov
- opis zahteve:
  - prioriteta
  - predlogi možnih rešitev
  - koristi (če je zahteva izpolnjena)
- gre za formalen opis zahtev: ne samo kot zabeležka v okviru intervjuja
- standarden obrazec



## SSADM: Analiza obstoječega stanja



**I5 - Planiranje analize**

- odločitev o področjih in metodah proučevanja
- podrobne zadolžitve posameznih članov delovne skupine
- vključitev uporabnikov: intervjuji, pregledovanje opravljnega dela, sodelovanje pri analizi (kot člani delovne skupine), anketiranje
- plan intervjujev:
  - zajeti različne uporabnike na različnih nivojih organizacije
  - dogovoriti se za natančen urnik

**I6 – Odobritev nadaljevanja**

- ta korak je zelo pomemben, zato je treba rezultate preveriti
- formalen pregled z najvišjimi predstavniki uporabnikov

## SSADM: Analiza obstoječega stanja



**Korak 120 - Raziskava in določitev zahtev (Investigate and Define Requirements)**

***Definicija:*** *Zahteva opisuje potrebno lastnost novega sistema*

**Poznamo:**

a) funkcionalne zahteve: kaj naj bi nov sistem delal

b) nefunkcionalne zahteve: kako dobro naj bi bila neka lastnost zagotovljena (stopnja kakovosti)

**Naloga tega koraka:** ugotoviti (elicit), definirati, prečistiti (refine) in dokumentirati zahteve v Seznamu zahtev (Requirements Catalogue)

- zadostuje doseči soglasje o zahtevah in njihovih prioritetah, da bi lahko izdelali različne opcije (different business options)
- v nadaljnjih korakih vsako zahtevo še bolj podrobno razdelamo (Requirements Specification)





## SSADM: Analiza obstoječega stanja



**Seznam zahtev uporabimo kot osnovo za:**

- preverjanje, kako predlagani sistem izpolnjuje zahteve
- primerjanje različnih možnih rešitev

**a) Funkcionalne zahteve**

- to so aktivnosti (funkcije), ki jih mora izvajati nov sistem, npr. zajem podatkov o vpisu, izpis pregleda opravljenih izpitov
- veliko funkcionalnih zahtev ugotovimo med modeliranjem podatkovnih tokov (bolj natančno: med izdelavo trenutnega fizičnega modela podatkovnih tokov)
- podrobneje jih razdelamo v okviru logičnega modela podatkovnih tokov
- posebno pozornost nameniti poizvedovanjem: le-ta običajno niso prikazana v diagramih podatkovnih tokov, zato jih je treba dokumentirati v seznamu zahtev

## SSADM: Analiza obstoječega stanja



**b) Nefunkcionalne zahteve**

- ostale zahteve, npr. performanse (odzivni čas), varnost in zaščita, ponovna vzpostavitev (okrevanje) v primeru okvare, arhiviranje, nadzor ipd.
- običajno so vezane na posamezne funkcionalne zahteve, npr. sistem mora izvesti poizvedovanje o razpoložljivih vozilih v 5 sekundah
- lahko pa se nanašajo na sistem kot celoto ali nek večji del sistema, npr. oprema mora biti robustna, da lahko obratuje tudi v garažah

**Vrste nefunkcionalnih zahtev:**

- **Zahteve, ki se nanašajo na nivo uslug** (Service Level Requirements) tj. na performanse novega sistema:
  - odzivni čas
  - razpoložljivost (v katerih obdobjih dneva je funkcija na razpolago)
  - zanesljivost (sprejemljiv čas, ko sistem ne obratuje (downtime); sprejemljivo število izpadov sistema ipd.)
- Te zahteve so pomembne v fazah 5 in 6





## SSADM: Analiza obstoječega stanja



- **Omejitve dostopa:**
  - kdo lahko dostopa do določenih podatkov: v tem koraku so samo grobo definirane, kasneje natančneje opredeljene (v opisu entitet, atri-butov in razmerij)
  - fizično varovanje
  - kodiranje podatkov (data encryption)
- **Ponovna vzpostavitev sistema (okrevanje, obnova):**
  - kompleksno vprašanje
  - zahteve vplivajo na izbor hardware-a in software-a
  - drage rešitve
  - proučitev rezervnih variant: začasni prehod na ročni sistem, uporaba neke druge instalacije

## SSADM: Analiza obstoječega stanja



- **Nadzor**
  - pregled nad dodeljevanjem funkcij posameznim uporabnikom
  - možnost sledenja
- **Omejitve (ki izvirajo iz okolja)**
  - zahteve za konverzijo (prehod) z obstoječega sistema na nov sistem
  - povezave z drugimi računalniškimi sistemi
  - posebne zahteve glede interakcije z uporabnikom (npr. invalidi)
  - vpliv okolja (temperatura, vlaga, umazanija)
- **Arhiviranje**
  - samo grob opis
  - podrobnejaša opredelitev ob razvoju zahtevanega logičnega podatkovnega modela in podrobnem opisu entitet



## SSADM: Analiza obstoječega stanja



**R1 - Seznam zahtev (Requirements Catalogue)**

- Pomen: da ne ostane kot neka zabeležka intervjuja ali stranski rezultat neke druge aktivnosti, ampak da so zahteve sistematično zbrane
- SSADM predlaga ustrezen obrazec, ki ga lahko projekt prilagodi svojim potrebam
- Seznam zahtev je najpomembnejši rezultat tega koraka

**R2 - Seznam uporabnikov**

- Opisuje uporabnike obstoječega sistema in njihove aktivnosti
- 2 stolpca: za vsako delovno mesto (job title) je podan opis aktivnosti
- Namen: osnova za opis vlog uporabnikov (user roles) v novem sistemu
- Povezava z organiziranostjo podjetja:
- Vzpostavitev ustrezne organizacije sicer ni neposredna naloga projekta, vendar je uspeh projekta odvisen tudi od ustrezne organizacije, zato je treba opozoriti na morebitne nepravilnosti

## Korak 130 - Raziskava obstoječih procesov (Investigate Current Processing)

**Namen:** Predstaviti delovanje obstoječege (fizičnega) sistema v obliki DFD, s katerimi soglašajo tudi uporabniki (dobiti soglasje uporabnikov)

**Splošno o vlogi DFD:**

- stična točka med razvijalci in uporabniki
- omogočajo natančen in jedrnat opis trenutnga sistema
- služijo kot osnova za nadaljnje načrtovanje
- izdelava DFD poteka v sodelovanju z uporabniki: analitik na podlagi intervjuja (in drugih informacij) izdela osnutek diagrama, ki ga dopolnjuje v razgovoru z uporabniki
- uporabniki morajo razumeti notacijo DFD: potrebno je izboraževanje





**Kako pričeti z risanjem DFD**

Zaradi velike količine informacij je na začetku včasih težko zajeti celoten sistem: možni so različni pristopi (izhodišča):

**a) Fizični tok dokumentov:** primeren, če se trenutni sistem sestoji predvsem iz tokov informacij v obliki dokumentov ali računalniških vhodov in izhodov (administrativni, ne pa proizvodni sistem)

**Koraki:**

1. korak: seznam dokumantov + ostalih informacijskih tokov (npr. informacije, posredovane po telefonu ali preko računalnika)
2. korak: risanje toka dokumentov
3. korak: določitev meje za analizo obstoječega sistema
4. korak: identifikacija procesov znotraj sistema

**b) Fizični tok dobrin (goods) – materialov in surovin:** podobno, primeren za proizvodne procese

## SSADM: Analiza obstoječega stanja



**c) Organizacijska struktura**

- izhodišče so procesi, ki jih izvajajo posamezne organizacijske enote
- začnemo z razmeroma enostavnim diagramom, v katerem vsaka organizacijska enota izvaja en proces, nato izvajamo dekompozicijo teh procesov in pripadajočih podatkovnih tokov
- ta pristop je primernejši za velike sisteme: tam lahko tokovi dokumentov vodijo do zelo kompliciranih diagramov

**Koraki:**

- izhodišče je organizacijska struktura
- uporabnik določi, katere organizacijske enote pridejo v poštev za analizo
- za vsako organizacijsko enoto predvidimo en proces kot osnovni proces na najvišjem nivoju
- dodamo informacijske tokove med procesi, procesi in zunanjimi entitetami ter procesi in podatkovnimi shrambami



## SSADM: Analiza obstoječega stanja



**Nekaj splošnih napotkov za dober DFD**

- idealnega DFD diagrama ni
- pomembno je, da se uporabnik in analitik strinjata, da diagram predstavlja razumen, smiselen (reasonable) in razumljiv (understandable) opis sistema
- izogniti se je treba prevelikim podrobnostim: paralysis by analysis

**Poenostavljanje preveč kompleksnih diagramov na nivoju 1:**

- kombiniranje procesov, dokler jih ne ostane samo 7+-2
- kombiniranje podatkovnih shramb
- kombiniranje zunanjih entitet
- podatkovni tokovi v podatkovne shrambe in iz njih: DFD kaže smer podatkovnega toka, ki je potreben za izvedbo procesa
- ažuriranja (updates) in poizvedovanja (retreivals): DFD prikazujejo le ažuriranja, poizvedovanja so dokumentirana v seznamu zahtev

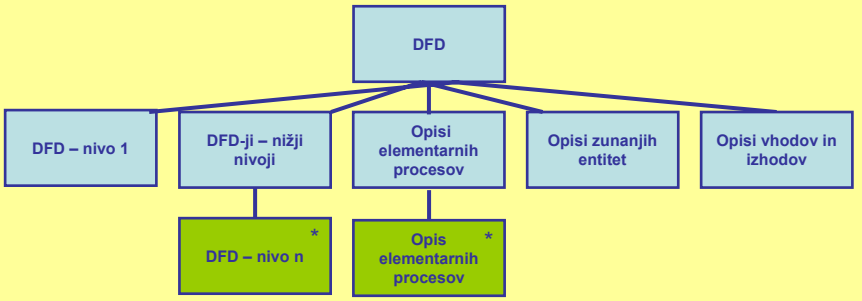
## SSADM: Analiza obstoječega stanja



**Dekompozicija procesov na nižje nivoje:**

- kadar je več kot 8 podatkovnih tokov v proces ali iz njega
- kadar je ime procesa kompleksno ali zelo splošno

**Sestavni deli modela DFD**






## SSADM: Analiza obstoječega stanja



**a) Opis elementarnih procesov:**

- opisno (narrative) ali:
  - psevdo koda
  - akcijski diagrami
  - strukturirana angleščina/slovenščina
  - odločitvene tabele
  - formalni jeziki za specifikacije

**b) Opis zunanjih entitet:** opisno (narrative)

- vloga
- omejitve
- funkcije zunajih entitet

c) **Opisi vhodov in izhodov:** vsebina podatkov, ki nastopajo v podatkovnih tokovih v sistem in iz njega na najnižjem nivoju (atributi)

**Korak 140 - Raziskava obstoječih podatkov (Investigate current data)**

**Cilj:**

- izdelati diagram LDS (logično podatkovno strukturo obstoječega sistema)
- paralelno s koraki 120 in 130

**Izdelki tega koraka:**

- Logični podatkovni model obstoječega sistema
  - diagram LDS (logična podatkovna struktura)
  - opis entitet
  - opis razmerij
- Podatkovni katalog





**Postopek izdelave logičnega podatkovnega modela:**

a) določitev (identifikacija) entitet

b) določitev razmerij

c) risanje diagrama (podrobna opredelitev razmerij)

d) izdelava spremljajoče dokumentacije

e) validacija logičnega podatkovnega modela

**a) Določitev (identifikacija) entitet**

Entiteta je nekaj,

- kar je pomembno za sistem
- o čemer lahko hranimo informacije
- kar lahko enolično določimo

**Vprašanje: ali so to entitete ali samo atributi (...)**

## SSADM: Analiza obstoječega stanja



**4 kriteriji, ki jih mora izpolnjevati entiteta:**

- mora biti pomembna za sistem, ki ga proučujemo
- z njo morajo biti povezani določeni podatki (mora imeti atribute)
- biti mora več konkretnih entitet istega tipa
- vsaka konkretna entiteta mora biti enolično določljiva (primarni ključ)

**b) Določitev razmerij**

- ugotavljanje, katere entitete so med seboj povezane
- matrika omogoča sistematičen pristop, hitrejše je risanje razmerij v diagramu

**Vprašanji, na katera iščemo odgovor:**

- Ali so (konkretne) entitete A neposredno povezane z entitetami B?
- Ali je ta neposredna povezava pomembna (zanimiva) za naš sistem?
- Včasih še: Ali je potreben dostop od ene entitete do druge?





## SSADM: Analiza obstoječega stanja



**c) Risanje diagrama (podrobna opredelitev razmerij)**

Narišemo posamezne entitete in razmerja med njimi

Pri vsakem razmerju podrobno opredelimo

- vrsto (stopnjo - degree) razmerja (1:1, 1:N, M:N)
- obveznost razmerja (ali je razmerje opcijsko ali obvezno)
- poimenovanje razmerja

Spremembe diagrama zaradi razmerij 1:1, M:N, ...

**d) Izdelava spremljajoče dokumentacije (SSADM obrazci ali CASE orodje):**

- opis entitet (obrazec)
- opis razmerij (obrazec, obe smeri razmerja opišemo)
- podatkovni katalog (ni sestavni del LDM: opis posameznih atributov in skupin domen)

## SSADM: Analiza obstoječega stanja



**Pomen atributov** (podatkovni element in atribut sta v SSADM sinonima)

- sestavljajo entitete
- nastopajo v procesih (v DFD diagramih) in v vhodno/izhodnih obrazcih
- podatkovni katalog vzpostavlja konsistentnost:
  - enotne definicije za celoten projekt
  - odprava homonimov in sinonimov

**Pomen skupinskih domen**

- gre za skupine sorodnih atributov z istim formatom, istim obsegom vrednosti, z istimi kontrolami

**e) Validacija logičnega podatkovnega modela**

- odprava redundantnih pristopnih poti
- podpora vsem procesom v obstoječem sistemu
- pregled z uporabnikom





## SSADM: Analiza obstoječega stanja



**Pojma navigacija (navigation) in pristopna pot (access path):**

- razmerja med entitetami omogočajo premikanje od ene entitete k drugi, pa čeprav je vmes več drugih entitet in razmerij
- temu premikanju po modelu rečemo navigacija
- pot, ki jo pri tem prehodimo, je pristopna pot

**Podpora vsem procesom:**

- navzkrižna kontrola z DFD (procesi za ažuriranje) in elementarnimi opisi procesov
- navzkrižna kontrola z Requirements Catalogue (poizvedovanja)
- pristopne poti morajo zagotavljati dostop do vseh potrebnih podatkov za izvršitev nekega procesa
- v tem koraku je validacija neformalna, v koraku 360 formalen opis v obliki Effect Corresponding Diagram (ažuriranja) in Enquiry Access Path (poizvedovanja)

## SSADM: Analiza obstoječega stanja



**Korak 150 -Oblikovanje logičnega pogleda na poslovanje (Derive logical view of services)**

**Predelava DFD iz koraka 130:**

- doslej: fizični opis dejanskega stanja (trenutni fizični DFD)
- v tem opisu dejanske funkcije sistema včasih zamegljene zaradi anomalij v sistemu: npr. podvajanje funkcij in podatkov
- sedaj: izoblikovati logični na delovanje sistema ne glede na dejansko implementacijo posmeznih funkcij

**Značilnosti logičnega pogleda na sistem:**

- vse informacije so v sistemu shranjene samo enkrat
- vsi procesi imajo dostop do teh informacij
- vsi podatkovni tokovi vsebujejo samo tiste podatke, ki se uporabljajo v pripadajočem procesu





## SSADM: Analiza obstoječega stanja



**Namen oblikovanja logičnega pogleda na sistem:**

- identifikacija problemov v obstoječem sistemu
- razumevanje osnovnih funkcij obstoječega sistema
- natančnejša opredelitev mej sistema
- preusmeritev pozornosti z obstoječega sistema na nov sistem
- vzpostavitev osnove za specifikacijo potrebnega sistema

**Koraki v oblikovanju logičnega pogleda na sistem:**

a) oblikovanje logičnih podatkovnih shramb

b) oblikovanje logičnih procesov in podatkovnih tokov na najnižjem nivoju

c) grupiranje procesov z nižjega nivoja v DFD diagrame na višjih nivojih

d) preverjanje konsistentnosti in popolnosti

e) dopolnitev in razširitev spremljajoče dokumentacije

## SSADM: Analiza obstoječega stanja



**Namen oblikovanja logičnega pogleda na sistem:**

- identifikacija problemov v obstoječem sistemu
- razumevanje osnovnih funkcij obstoječega sistema
- natančnejša opredelitev mej sistema
- preusmeritev pozornosti z obstoječega sistema na nov sistem
- vzpostavitev osnove za specifikacijo potrebnega sistema

**Koraki v oblikovanju logičnega pogleda na sistem:**

a) oblikovanje logičnih podatkovnih shramb

b) oblikovanje logičnih procesov in podatkovnih tokov na najnižjem nivoju

c) grupiranje procesov z nižjega nivoja v DFD diagrame na višjih nivojih

d) preverjanje konsistentnosti in popolnosti

e) dopolnitev in razširitev spremljajoče dokumentacije




## SSADM: Analiza obstoječega stanja



**a) Oblikovanje logičnih podatkovnih shramb**

- poteka v povezavi z entitetami v logičnem podatkovnem modelu
- vsaka podatkovna shramba predstavlja eno ali več entitet
- vsaka entiteta pripada samo eni podatkovni shrambi

**b) Oblikovanje logičnih procesov in podatkovnih tokov na najnižjem nivoju**

- logični proces: proces, ki preoblikuje ali uporablja podatke, ker to zahteva poslovanje sistema; je neodvisen od implementacije
- pri vsakem procesu odpade opis, vezan na implementacijo, npr. "izpolni obrazec ta in ta" se spremeni v "zabeleži podatke ..."
- ločiti ročne procese od računalniško podprtih: včasih problem, npr. avtorizacija
- ročni procesi se ne prikazujejo na DFD, razen v primeru, ko je tudi ročni sistem predmet pogodbe
- odpade opis lokacije, kjer se proces izvaja

## SSADM: Analiza obstoječega stanja



**c) Grupiranje procesov z nižjega nivoja v DFD diagrame na višjih nivojih**

- logične podatkovne shrambe in procese vpeljemo najprej na najnižjih nivojih
- sledi grupiranje (oblikovanje višjih nivojev)
- pogosto izvedemo ločitev na 2 vrsti procesov:
  - procesi, potrebni za izvajanje poslovanja (npr. izstavljanje faktur)
  - procesi za vzdrževanje podatkov, ki podpirajo poslovanje (npr. o kupcih)

**d) Preverjanje konsistentnosti in popolnosti**

- primerjava s trenutnim fizičnim opisom podatkovnih tokov: upoštevani morajo biti vsi obstoječi procesi, podatkovni tokovi in podatkovne shrambe
- kontrola smiselnosti (common-sense checks): ali se prikazani procesi skladajo s predstavo analitika o delovanju sistema
- potencialne napake: pozorno pregledati procese in podatkovne shrambe, v katere bodisi samo vstopajo ali samo izstopajo podatkovni tokovi
- preverjanje z uporabniki: ali so zajete vse funkcije sistema





## SSADM: Analiza obstoječega stanja



**e) Dopolnitev in razširitev spremljajoče dokumentacije**

- dokumentacija je bila izdelana v sklopu analize obstoječega sistema
- sedaj: dopolniti opis elementarnih procesov, zunanjih entitet, vhodov in izhodov v skladu z logičnimi DFD
- navzkrižna kontrola opisa el.procesov z možnostjo navigacije v LDS

**Korak 160 - Združitev rezultatov (Assemble investigation results)**

**Splošno:**

- zadnji korak vsake faze
- oblikujemo dokumente, ki jih je treba formalno pregledati in ki predstavljajo osnovo za naslednjo fazo

## SSADM: Analiza obstoječega stanja



**e) Izdelki faze 1: Analiza obstoječega stanja**

- seznam zahtev
- opis trenutnega poslovanja/stanja (Current services description): slika 3.54
- podatkovni katalog
- logični podatkovni model trenutnega okolja
- kontekstni diagram
- logični model podatkovnih tokov
- navzkrižni diagram podatkovnih shramb in entitet



## SSADM: Opcije poslovnega sistema

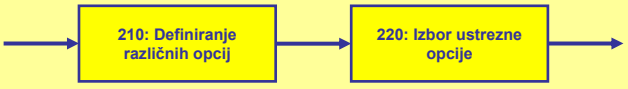


**Faza 2: Opcije poslovnega sistema (Business System Options)**

- **Izhodišče:** zahteve glede novega sistema lahko izpolnimo na več načinov
- Predstavimo več možnih rešitev, vodstvo izbere tisto, ki služi kot osnova za nadaljnje delo
- Dokončni prelom z obstoječim sistemom: usmeritev na nov sistem
- Kaj naj nov sistem dela, ne pa kako

**2 koraka:**

- opcije najprej definiramo (korak 210)
- nato naročnik izbere ustrezno opcijo (korak 220)



## SSADM: Opcije poslovnega sistema



**Korak 210 - Definiranje različnih opcij (Define Business System Options)**

**Vhodi:**

- Seznam zahtev (glavni vhod)
- Logični model podatkovnih tokov
- Dokument o vzpostavitvi projekta
- Logični podatkovni model: samo če opcije zahtevajo spremembo logičnega podatkovnega modela

**Izhodi:** 2-6 opcij, vsaka opcija vsebuje:

- **opis funkcij:** tekstualni opis, po potrebi podprt z DFD na nivoju 1 in LDS, meje sistema, osnovni tehnični vidiki (samo v smislu vpliva na poslovanje), roki za izdelavo
- **analizo stroškov in koristi:** stroški razvoja, stroški obratovanja, pričakovane koristi - neposredne, posredne (tangible/intangible)
- **analizo vpliva (impact analysis):** spremembe v organizaciji, spremembe v načinu dela uporabnikov, potrebe po izobraževanju



## SSADM: Opcije poslovnega sistema



**Pristop:**

- dogovor o minimalnem naboru zahtev, ki jih mora izpolnjevati vsaka opcija (na podlagi seznama zahtev in dokumenta o vzpostavitvi projekta)
- kreativno razmišljanje
- v skupinski diskusiji analizirati čim več idej
- brainstorming
- vključevanje uporabnikov
- za majhne projekte zadostujeta 2 opciji: minimalna in maksimalna
- minimalna: osnovna rešitev za zahteve z najvišjo prioriteto
- maksimalna: kompleksna rešitev za vse zahteve
- izbrana rešitev lahko združuje lastnosti večjega števila predlaganih opcij
- vedno je možna tudi opcija, da obdržimo obstoječi sistem (mogoče z manjšimi dopolnitvami)

## SSADM: Opcije poslovnega sistema



**Različne opcije oblikujemo glede na:**

a) porazdelitev sistema

- zahteva ustrezne organizacijske spremembe
- proučiti je treba vpliv organizacijskih sprememb (tveganje)

b) meje sistema

- z različnimi opcijami postavljamo meje sistema za načrtovanje, ne za analizo
- znotraj mej sistema je treba definirati še meje med človekom (uporabnikom) in sistemom

c) stopnja avtomatizacije

- Primer za stopnjo avtomatizacije: vodenje skladišča
  - tedenski izpis stanja zalog
  - vsakodnevni izpis izdelkov s kritično zalogo
  - avtomatsko generiranje naročila, ko zaloga pade pod predpisano mejo



## SSADM: Opcije poslovnega sistema



**Korak 220 - Izbor ustrezne opcije (Select Business System Option)**

**Predstavitev opcij:**

- pri majhnih projektih zadostujejo neformalni razgovori z uporabniki
- pri velikih projektih formalna predstavitev pred ustreznimi organi (projektni svet: najvišje vodstvo, predstavniki uporabnikov, predstavniki AOP pri naročniku, neodvisni ocenjevalci)

Naročnik izbere ustrezno opcijo ali kombinirano rešitev, ki združuje lastnosti več opcij

Izbrana opcija predstavlja osnovo za naslednji modul: specifikacija zahtev

## Splošno o specifikaciji zahtev

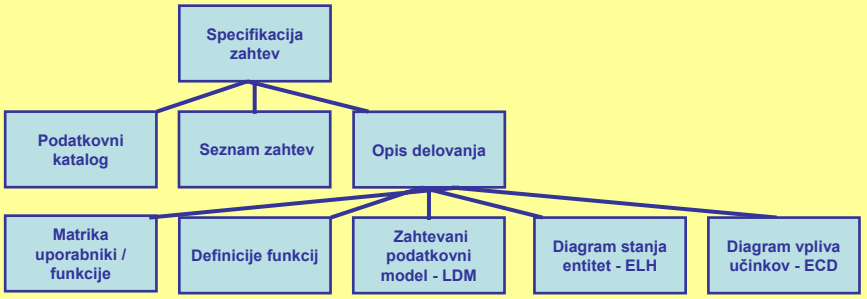
- requirements specification
- zaenkrat imamo samo začetni načrt izbrane opcije poslovnega sistema
- ta načrt je treba pretvoriti v podrobno logično specifikacijo novega sistema
- ta modul vsebuje samo eno fazo, ki pa je najbolj kompleksna (tako po strukturi kot po uporabljenih tehnikah)
- rezultat je dokument Specifikacija zahtev: odraža logični pogled na nov sistem, čeprav je ta pogled le redko v celoti neodvisen od konkretne implementacije





## 2. Faza 3: Specifikacija zahtev

Rezultati



## SSADM: Specifikacija zahtev



**Rezultati**

- **Podatkovni katalog, seznam zahtev:** dopolnitev že obstoječih
- **Matrika uporabniki/funkcije:**
  - povezava med uporabniki in funkcijami (kateri uporabniki imajo neposreden dostop do katerih funkcij)
  - osnova za definicijo menujev
- **Definicije funkcij:** opisujejo procese v novem sistemu
  - **I/O strukture:**
    - struktura in vsebina informacij, ki vstopajo v sistem in iz njega
  - **Pristopne poti pri povpraševanju (Enquiry Access Paths – AEP):**
    - navigacija po podatkovnem modelu pri poizvedbah
  - **Opisi elementarnih in skupnih procesov:**
    - opis podrobnosti pri posameznih procesih (iz podatkovnih tokov)



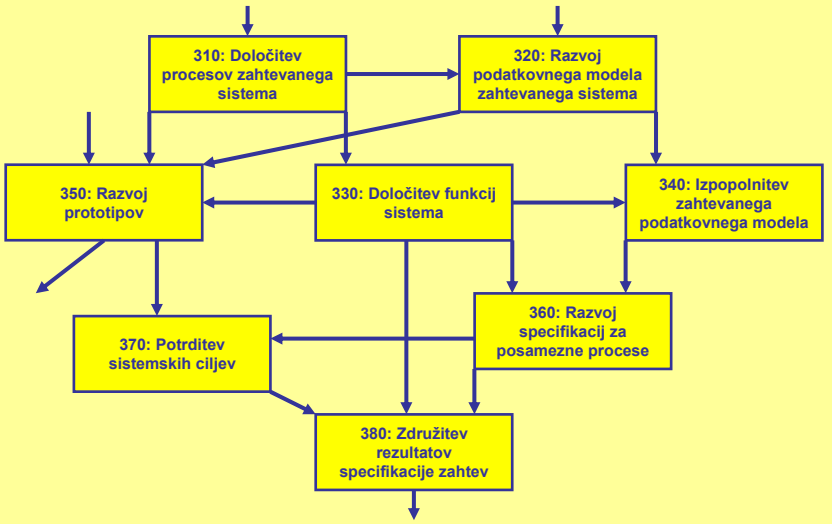
## SSADM: Specifikacija zahtev



- **Zahtevani sistemski podatkovni model - LDM:**
  - **Diagram LDS**
  - **Opis entitet**
  - **Opis razmerij po relacijski analizi**
- **Diagram stanja entitet - ELH:**
  - prikaz dogodkov, ki učinkujejo na vsako entiteto
  - za vsak dogodek so prikazane operacije nad podatki
- **Diagrami vpliva učinkov (Effect Correspondence Diagrams – ECD):**
  - vpliv dogodka na vse entitete v modelu
  - nasprotje od ELH: tam ena entiteta in vsi dogodki, tu: en dogodek in vse entitete, na katere učinkuje

## SSADM: Specifikacija zahtev









## SSADM: Specifikacija zahtev



**Pregled korakov**

- za izbrano opcijo še nimamo DFM in LDM modela: v 310 in 320 izdelamo ta modela, izhajajoč iz logičnega DFM modela (150) in trenutnega LDM (140)

**310 Določitev procesov zahtevanega sistema - Define Required System Processing**

- izdelava modela podatkovnih tokov (DFM) za zahtevani sistem
- vključuje DFD, opise zunanjih entitet, opise elementarnih procesov in opise I/O podatkov
- zunanje entitete sovpadajo z uporabniki novega sistema

**320 Razvoj podatkovnega modela zahtevanega sistema - Develop Required Data Model**

- vključitev novih zahtev v podatkovni model
- dopolnitev opisov entitet: sedaj zajeti vsi atributi, vse nefunkcionalne zahteve (dostop, varnost, arhiviranje ...)

## SSADM: Specifikacija zahtev



**330 Določitev funkcij sistema - Derive System Functions**

- določitev funkcij kot osnovnih enot procesiranja:
- funkcije ažuriranja: z grupiranjem elementarnih procesov iz DFD
- funkcije poizvedovaja: na podlagi seznama zahtev
- I/O strukture: struktura vhodov in izhodov (kot pričetek dokumentacije, povezane s posameznimi funkcijami)

**340 Izpopolnitev zahtevanega podatkovnega modela - Enhance Required Data Model**

- uporaba relacijske analize na I/O strukturah: bottom-up pristop, ki upošteva funkcionalne odvisnosti med atributi (normalizacija)

**350 Razvoj prototipov - Develop Specification Prototypes**

- neobvezen korak
- izdelava prototipov za tiste dele aplikacije, ki so kritični ali jih uporabnik ne zna natančno definirati





## SSADM: Specifikacija zahtev



**360 razvoj specifikacij za posamezne procese - Develop Processing Specification**

- identifikacija dogodkov (iz definicije funkcij)
- izdelava ELH diagramov (diagrami sprememb stanja entitet)
- izdelava Effect Corresponding Diagrams za vsak dogodek (diagram učinkov)
- izdelava Enquiry Access Paths (pristopne poti za kompleksna poizvedovanja)

**370 potrditev sistemskih ciljev - Confirm System Objectives**

- pregled seznama zahtev s ciljem ugotoviti, ali so bile vse zahteve upoštevane

**380 Združitev rezultatov specifikacije zahtev -Assemble Requirements Specification**

- združitev rezultatov v zaključni dokument
- kontrola konsistentnosti
- formalna kontrola kakovosti (quality assurance review)

## SSADM: Specifikacija zahtev



**Določitev procesov zahtevanega sistema (Define Required System Processing) - korak 310**

**Cilj koraka:** določiti procese (postopke) zahtevanega sistema s pomočjo modela podatkovnih tokov (DFM)

**Opis procesov je podan z:**

- množico DFD diagramov
- I/O opisi (opisi vsebine podatkovnih tokov, ki prečkajo mejo sistema)
- opisi zunanjih entitet in uporabnikov novega sistema
- podrobnimi opisi elementarnih procesov (tj. procesov na najnižjem nivoju)





## SSADM: Specifikacija zahtev



**Pristop:**

- izhodišče predstavlja izbrana opcija
- izdelava diagramov podatkovnih tokov za zahtevani sistem
- opis elementarnih procesov
- opis vhodov in izhodov (I/O Descriptions)
- opis vsebine podatkovnih tokov v sistem in iz njega gledano na najnižjem nivoju (od zunanjih entitet v sistem in obratno)
- opis zunanjih entitet
- seznam uporabnikov in vloge uporabnikov (user roles)

## SSADM: Specifikacija zahtev



**Razvoj podatkovnega modela za zahtevani sistem (Develop required data model) - korak 320**

- korak 140: podatkovni model obstoječega sistema
- izbrana opcija poslovnega sistema prinaša dodatne zahteve: te zahteve je treba vgraditi v podatkovni model
- dopolnitve se odražajo na več načinov:
  - nove strukture (entitete in razmerja)
  - dopolnitev spremljajoče dokumentacije (predvsem nefunkcionalne zahteve, kot npr. pravica do do-stopa, zaščita, arhiviranje)
- Validacija





## SSADM: Specifikacija zahtev



**Določitev funkcij sistema (Derive System Functions) - korak 330**

**Pojem funkcije:**

- zaključena enota procesiranja, ki se izvrši naenkrat, da bi podprla neko poslovno aktivnost
- v SSADM osnovni gradnik pri opisu procesiranja: na funkcije se navezuje vsa ostala dokumentacija, ki se nanaša na procesiranje
- naivna predstava: analogija z opcijo v menuju

## SSADM: Specifikacija zahtev



**Univerzalni model funkcije:**

- konceptualni prikaz zaradi razumevanja vseh komponent
- vsak vhod se preveri, povzroči nato ažuriranje ali poizvedbo ter na koncu izpis rezultatov (in morebitnega sporočila o napakah)
- opis vhodov in izhodov: I/O Structures
- opis ažuriranj: dogodki, Effect Correspondence Diagrams, kasneje Update Process Models
- opis poizvedovanj: Enquiry Access Paths, kasneje Enquiry Process Models





## SSADM: Specifikacija zahtev



**Pristop v tem koraku:**

- identifikacija funkcij
- identifikacija dogodkov (in povezava dogodkov s funkcijami)
- začetna identifikacija dialogov
- dokumentiranje funkcij
- opis I/O struktur

**Trije kriteriji za klasifikacijo:**

- iniciator funkcije: uporabnik ali sistem
- vpliv funkcije na podatke: ažuriranje ali poizvedba
- način delovanja: on-line ali off-line

## SSADM: Specifikacija zahtev



**Funkcije, ki jih sproži uporabnik:**

- uporabnik se odloči, da bo izvedel neko funkcijo
- izbor opcije v menuju, klik z miško
- v Required DFD: podatkovni tok od zunanje entitete v sistem
- poizvedovanja: v seznamu zahtev

**Funkcije, ki jih sproži sistem:**

- računalniški sistem sam (avtomatsko) sprejme odločitev, da izvede neko funkcijo
- ta odločitev je lahko pogojena časovno ali nastopi kot posledica izvršitve nekega drugega procesa
- Required DFD: podatkovni tok iz procesa v podatkovno shrambo (ni vhodnega podatkovnega toka)
- poizvedovanja: seznam zahtev, tok iz sistema



## SSADM: Specifikacija zahtev



**Funkcije, ki povzročijo ažuriranje podatkov:**

- spreminjajo vrednosti v podatkovni bazi
- Required DFD: podatkovni tok iz procesa v podatkovno shrambo

**Funkcije, ki realizirajo poizvedovanja:**

- samo posredovanje podatkov iz podatkovne baze uporabnikom ali drugim sistemom
- vir: seznam zahtev
- nekatera ažuriranja vključujejo manjša poizvedo-vanja: takih poizvedovanj ne formuliramo kot samostojne funkcije

## SSADM: Specifikacija zahtev



**On-line funkcije:**

- interaktivno izvajanje v obliki dialoga med uporabnikom in računalnikom
- tipično za današnje sisteme
- kompleksno načrtovanje vnosa podatkov, lažja obravnava napak

**Off-line funkcije:**

- ko se funkcija sproži, poteka (se izvede) brez intervencije uporabnika
- enostavno načrtovanje vhodnih podatkov, problemi pri obdelavi napak
- v poštev pride:
  - ko je treba obdelati veliko število enakih ali podobnih transakcij
  - ko so vhodni podatki podani v računalniški obliki
  - premajhne računalniške zmogljivosti: zajem transakcij on-line, ažuriranje ponoči off-line
- odločitev za on-line ali off-line: v Poslovnih opcijah sistema



## SSADM: Specifikacija zahtev



**Izpopolnitev zahtevanega podatkovnega modela (Enhance Required Data Model) - korak 340**

Dosedanji pristop k izdelavi podatkovnega modela je bil **top-down:**

- določitev entitetnih tipov (osnovnih nosilcev podatkov)
- določitev razmerij med entitetami
- določitev (najpomembnejših) atributov za vsako entitetni tip
- validacija: model mora podpirati predvidene procese

**Pomanjkljivosti:**

- za vsak entitetni tip smo določili samo najpomembnejše atribute
- dodeljevanje atributov posameznim entitetam je bilo subjektivno (ni temeljilo na uporabi neke formalne tehnike)
- niso bila upoštevana razmerja (odvisnosti) med atributi, ki so lahko pomembna pri posmeznih poizvedovanjih

## SSADM: Specifikacija zahtev



**Cilj: izdelati podroben načrt podatkov,**

- ki ga bo moč pretvoriti v fizični opis podatkovne baze
- ki bo fleksibilen (v smislu kasnejšega dodajanja novih zahtev)
- ki bo minimalen (neredundanten)

**Z drugimi besedami: v smislu relacijskih podatkovnih baz**

- želimo natančno definirati relacijske sheme posameznih relacij
- vsaka relacija ima primarni ključ, ki enolično določa vrednosti ostalih atributov





## SSADM: Specifikacija zahtev



**Pristop SSADM: Relacijska analiza podatkov (Relational data analysis)**

- za izhodišče vzamemo opise I/O struktur iz prejšnjega koraka (tu so natančno definirani atributi)
- z analizo ti. funkcionalnih odvisnosti med atributi grupiramo atribute v ti. normalizirane relacije (temu postopku rečemo preprosto normalizacija)
- združimo normalizirane relacije za vse I/O strukture (racionalizacija relacij)
- normalizirane relacije narišemo v obliki diagrama LDS
- primerjamo dobljen diagram LDS z diagramom, ki smo ga dobili po metodi top-down v koraku 320
- odpravimo morebitne konflikte in ažuriramo spremljajočo dokumentacijo

## SSADM: Specifikacija zahtev



**Namen RA :** grupirati vse atribute v množico dobro normaliziranih relacij

**Lastnosti (dobro) normaliziranih relacij:**

- ni nepotrebnega podvajanja atributov v različnih relacijah
- ni anomalij pri ažuriranju
- grupiranje atributov je stabilno (se ne spremeni ob dodajanju novih aplikacij)
- z drugimi besedami: minimalen, neredundanten, stabilen podatkovni model

**Postopek normalizacije:**

- več normalnih oblik (NF)
- SSADM obravnava 1., 2. in 3. normalno obliko
- obstajajo še Boyce-Coddova, 4. in 5. normalna oblika
- velika večina relacij, ki so v 3. NF, je tudi v v vseh nadaljnjih oblikah
- mi bomo obravnavali "inženirski" pristop; strogo teoretičen pristop v PB1
- postopek normalizacije bomo prikazali na primeru TabelaZdravil




## SSADM: Specifikacija zahtev



**a) Nenormalizirani podatki: na začetku**

- podatkov ne moremo predstaviti v tabelarični obliki
- ovira: ponavljajoča se skupina (tj. skupina atributov, ki se ponavljajo za isto vrednost primarnega ključa)

**Primer:** primarni ključ je številka pacienta

| Številka pacienta | Priimek pacienta | Ime pacienta | Številka oddelka | Naziv oddelka | Datum recepta | Šifra zdravila | Naziv zdravila | Doza | Čas jemanja |
|---|---|---|---|---|---|---|---|---|---|
| 923 | Novak | Jože | 10 | P.Držaja | 20.5.99 | C122 | Penicilin | 2x1 | 14 |
| | | | | | 24.5.99 | A51 | Apaurin | 1x1 | 5 |
| | | | | | 29.5.99 | A51 | Apaurin | 1x2 | 7 |
| 988 | Kebe | Andreja | 11 | UKC Lj | 1.6.99 | C110 | Morfij | 3x3 | 3 |
| | | | | | 1.6.99 | C122 | Penicilin | 3x2 | 7 |

**b) Prva normalna oblika**

- odpravimo ponavljajočo se skupino
- dobimo 2 relaciji:
  - prvo relacijo sestavljajo atributi, ki niso v ponavljajoči se skupini
  - drugo relacijo sestavljajo atributi ponavljajoče se skupine
- primarni ključ prve relacije je nespremenjen
- primarni ključ druge relacije je sestavljen iz:
  - ključa prve relacije (številka pacienta)
  - ključa ponavljajoče se skupine (datum+šifra zdravila)

**Primer: ...**





| Številka pacienta | Priimek pacienta | Ime pacienta | Številka oddelka | Naziv oddelka |
|---|---|---|---|---|
| 923 | Novak | Jože | 10 | P.Držaja |
| 988 | Kebe | Andreja | 11 | UKC Lj |

| Številka pacienta | Datum recepta | Šifra zdravila | Naziv zdravila | Doza | Čas jemanja |
|---|---|---|---|---|---|
| 923 | 20.5.99 | C122 | Penicilin | 2x1 | 14 |
| 923 | 20.5.99 | A51 | Apaurin | 1x1 | 5 |
| 923 | 25.5.99 | A51 | Apaurin | 1x2 | 7 |
| 988 | 1.6.99 | C110 | Morfij | 3x3 | 3 |
| 988 | 1.6.99 | C122 | Penicilin | 3x2 | 7 |

**c) Druga normalna oblika**

- odpravimo nepopolne odvisnosti od sestavljenega ključa (to pomeni, da moramo pregledati samo tiste relacije, ki imajo sestavljen ključ)
- atribute, ki niso v celoti odvisni od sestavljenega ključa, prestavimo v posebno relacijo

**Definicija funkcionalne odvisnosti:**

*atribut B je funkcionalno odvisen od atributa A, če vsaki vrednosti atributa A pripada ena sama vrednost atributa B*

Preprosteje rečemo, da A določa B

**Primer:** ime zdravila je odvisno samo od šifre zdravila, ne pa od celotnega sestavljenega ključa





| Številka pacienta | Priimek pacienta | Ime pacienta | Številka oddelka | Naziv oddelka |
|---|---|---|---|---|
| 923 | Novak | Jože | 10 | P.Držaja |
| 988 | Kebe | Andreja | 11 | UKC Lj |

| Številka pacienta | Datum recepta | Šifra zdravila | Doza | Čas jemanja |
|---|---|---|---|---|
| 923 | 20.5.99 | C122 | 2x1 | 14 |
| 923 | 20.5.99 | A51 | 1x1 | 5 |
| 923 | 25.5.99 | A51 | 1x2 | 7 |
| 988 | 1.6.99 | C110 | 3x3 | 3 |
| 988 | 1.6.99 | C122 | 3x2 | 7 |

| Šifra zdravila | Naziv zdravila |
|---|---|
| C122 | Penicilin |
| A51 | Apaurin |
| C110 | Morfij |

**d) Tretja normalna oblika**

- odpravimo tranzitivne odvisnosti
- vse atribute, ki niso neposredno odvisni od primarnega ključa, prestavimo v posebno relacijo

**Primer:** ime oddelka ni neposredno odvisno od številke pacienta, ampak od številke oddelka

**Notacija SSADM:**

- atribute pišemo vertikalno
- primarni ključ podčrtamo
- tuj ključ označimo z zvezdico (velja za 3. NF)

**Primer:** ...





| Številka pacienta | Priimek pacienta | Ime pacienta | Številka oddelka |
|---|---|---|---|
| 923 | Novak | Jože | 10 |
| 988 | Kebe | Andreja | 11 |

| Številka oddelka | Naziv oddelka |
|---|---|
| 10 | P.Držaja |
| 11 | UKC Lj |

| Številka pacienta | Datum recepta | Šifra zdravila | Doza | Čas jemanja |
|---|---|---|---|---|
| 923 | 20.5.99 | C122 | 2x1 | 14 |
| 923 | 20.5.99 | A51 | 1x1 | 5 |
| 923 | 25.5.99 | A51 | 1x2 | 7 |
| 988 | 1.6.99 | C110 | 3x3 | 3 |
| 988 | 1.6.99 | C122 | 3x2 | 7 |

| Šifra zdravila | Naziv zdravila |
|---|---|
| C122 | Penicilin |
| A51 | Apaurin |
| C110 | Morfij |

| UNF | 1 NF | 2 NF | 3 NF |
|---|---|---|---|
| Številka pacienta<br>Priimek pacienta<br>Ime pacienta<br>Številka oddelka<br>Naziv oddelka<br>Datum recepta<br>Šifra zdravila<br>Naziv zdravila<br>Doza<br>Čas jemanja | Številka pacienta<br>Priimek pacienta<br>Ime pacienta<br>Številka oddelka<br>Naziv oddelka<br><br>Številka pacienta<br>Datum recepta<br>Šifra zdravila<br>Naziv zdravila<br>Doza<br>Čas jemanja | Številka pacienta<br>Priimek pacienta<br>Ime pacienta<br>Številka oddelka<br>Naziv oddelka<br><br>Številka pacienta<br>Datum recepta<br>Šifra zdravila<br>Doza<br>Čas jemanja<br><br>Šifra zdravila<br>Naziv zdravila | Številka pacienta<br>Priimek pacienta<br>Ime pacienta<br>*Številka oddelka<br><br>Številka oddelka<br>Naziv oddelka<br><br>Številka pacienta<br>Datum recepta<br>Šifra zdravila<br>Doza<br>Čas jemanja<br><br>Šifra zdravila<br>Naziv zdravila |





**e) Preverjanje, če so relacije normalizirane**

**Test 1:** Ali je za eno vrednost primarnega ključa ena vrednost vsakega atributa?

**Test 2:** Ali je vsak atribut neposredno in v celoti odvisen od primarnega ključa?

**f) Združitev normaliziranih relacij za vse I/O strukture (racionalizacija relacij)**

- za vsako I/O strukturo dobimo množico normaliziranih relacij
- združimo relacije, ki imajo isti primarni ključ
- vsaki relaciji damo ime

**Hierarhični ključ:** sestavljen ključ, v katerem eden ali več delov nima enoličnega pomena (ni uporabljen kot primarni ključ v neki drugi relaciji)

**opredelitveni element** (qualifying element): atribut, ki je potreben, da postane kombinacija enolična

**Izdelava diagrama LDS za normalizirane relacije:**

3 pravila, pravilo 1 ima dve podpravili

**Pravilo 1:** Relacije prikažemo kot entitetne tipe

**Pravilo 1a:** Če del sestavljenega ključa ne obstaja kot primarni ključ v neki drugi relaciji, se prikaže kot entitetni tip in se označi kot tuj ključ v vseh relacijah, kjer nastopa.

**Pravilo 1b:** Če je sestavljen ključ hierarhičen, se opredelitveni atribut označi kot tuj ključ v tisti relaciji in v vseh drugih relacijah, kjer nastopa. Za neenolične atribute hierarhičnega ključa ne kreiramo posebnih entitetnih tipov.

**Pravilo 2:** Relacije s sestavljenim ključem so last relacij, ki imajo za primarni ključ posamezne dele sestavljenega ključa (majhni ključi si lastijo velike)

**Pravilo 3:** Relacije s tujim ključem so last relacij, v katerih je ta (tj. tuj) ključ primarni ključ. (vranja noga prijemlje zvezdice)

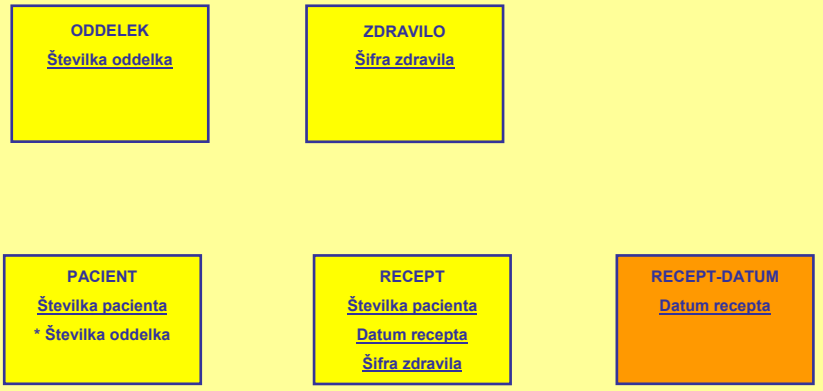
 



**Pravilo 1:** Relacije prikažemo kot entitetne tipe

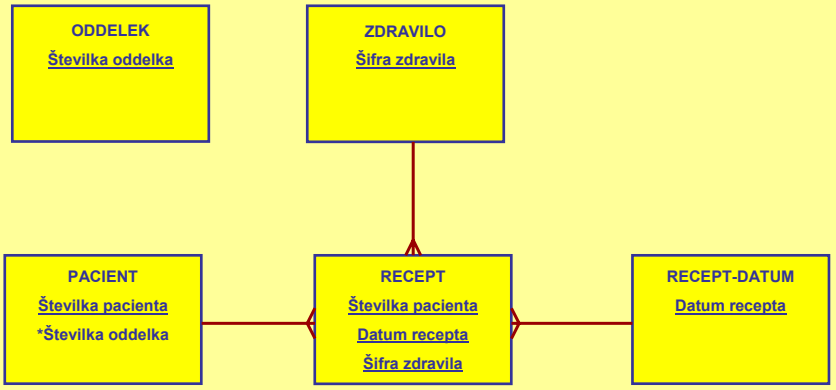
**Pravilo 1a:** Če del sestavljenega ključa ne obstaja kot primarni ključ v neki drugi relaciji, se prikaže kot entitetni tip in se označi kot tuj ključ v vseh relacijah, kjer nastopa.



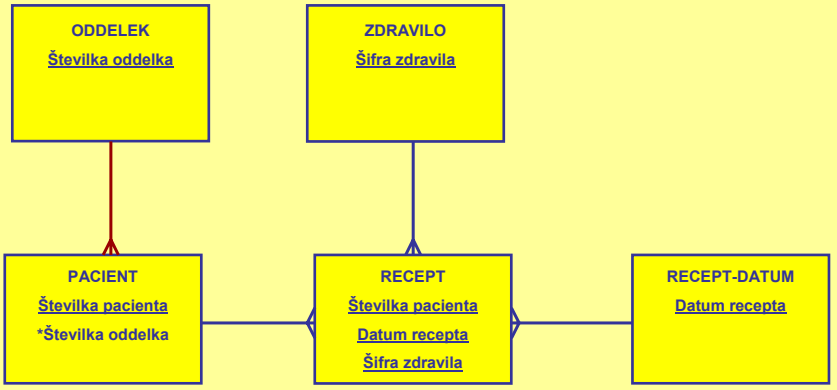


**Pravilo 2:** Relacije s sestavljenim ključem so last relacij, ki imajo za primarni ključ posamezne dele sestavljenega ključa (majhni ključi si lastijo velike)



**Pravilo 3:** Relacije s tujim ključem so last relacij, v katerih je ta (tj. tuj) ključ primarni ključ. (vranja noga prijemlje zvezdice)





**Primer:** Potrdilo o opravljenih izpitih
(relacijska analiza od obrazca do relacijske sheme)

**Različni pristopi k relacijski analizi**

- relacijsko analizo lahko delamo kadarkoli, tudi v fazi analize zahtev
- kot osnova lahko služijo (namesto I/O struktur in njihovih opisov):
  - obrazci, ki se uporabljajo v obstoječem sistemu
  - poročila
  - datoteke obstoječega sistema
  - opisi entitet (s tem, da je seznam atributov še nepopoln)
- nekateri projekti uporabljajo samo top-down ali samo bottom-up pristop
- v SSADM se oba pristopa dopolnjujeta

## SSADM: Specifikacija zahtev



**Razvoj prototipov (Develop specification prototypes) - korak 350**

**Osnovni namen:** pomoč pri specifikaciji zahtev, ker:

- uporabnik ne zna dovolj natančno opisati zahtev
- sodobna orodja omogočajo prototipiranje

**Dva osnovna tipa prototipov:**

- začasni prototipi ("thrown away")
- prototipi, ki jih razvijemo v končen izdelek

**SSADM pristop:**

- obseg prototipiranja je natančno določen: korak lahko tudi izpustimo
- natančno določen postopek izdelave, prikaz uporabniku, kriteriji za ponavljanje
- uporaba začasnih prototipov, razlikovanje od "rapid", "evolutionary" in "incremental prototyping"
- strikten nadzor vodstva projekta

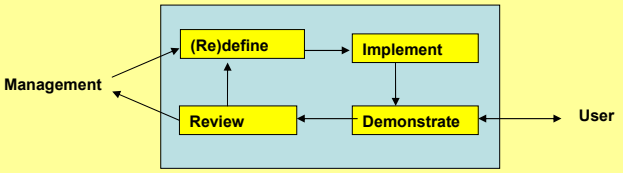


## SSADM: Specifikacija zahtev



**Tipičen postopek izdelave prototipa:**

- (ponovno) definiraj prototip
- izgradi prototip
- prikaži prototip
- obnovi prototip



## SSADM: Specifikacija zahtev



**Cilj prototipiranja:**

- preveriti tehnično izvedljivost specifikacij
- preveriti pravilnost specifikacij

**Težave pri uporabi prototipov:**

- uporabnik dobi občutek, da je mogoče celotno aplikacijo izdelati zelo hitro
- nekatera orodja za prototipiranje omogočajo izdelavo drugačnih (lepših) uporabniških vmesnikov, kot bodo v končni aplikaciji
- uporabnik dobi napačen občutek o hitrosti delovanja
- uporabnik želi spremeniti meje sistema
- problem "popolnega" prototipa
- pomanjkljiva dokumentacija prototipa
- nestandardni dialogi v prototipu





## SSADM: Specifikacija zahtev



**Razvoj specifikacij za posamezne procese – korak 360**

**Vsebina:** Podrobna specifikacija vseh potrebnih procesov

**Obsega:**

- izdelava diagramov ELH (vse spremembe stanja neke entitete; kako različni dogodki učinkujejo na eno entiteto)
- izdelava diagramov ECD (Effect Correspondence Diagrams): kako nek dogodek učinkuje na različne entitete
- izdelava diagramov EAP (Enquiry Access Paths): prikaz navigacije po podatkovnem modelu pri poizvedovanjih

## SSADM: Specifikacija zahtev



**Izdelava diagramov ELH**

- izdelava matrike Dogodki/Entitete (začetni seznam dogodkov je bil narejen v koraku 330)
- izdelava začetnih (initial, simple) diagramov ELH za vse entitete (najprej za "detail", nato za "master" entitete)
- izdelava dokončnih (final, full) diagramov ELH (najprej za "master", nato za "detail" entitete)
- dodajanje operacij
- dodajanje indikatorjev stanj (v koraku 520 Define Update Processes)





## SSADM: Specifikacija zahtev



**Izdelava diagramov ECD**

- na osnovi ELH diagramov izdelamo ECD diagram za vsak dogodek posebej
- uporabimo prilagojene Jacksonove diagrame
- tipično enostavni diagrami, le eden (nekaj) kompleksnih (kritičen dogodek)

**8 korakov izdelave ECD:**

- nariši pravokotnik za vsako entiteto, na katero vpliva izbran dogodek
- nariši dodatne pravokotnike za sočasne učinke
- dodaj pravokotnike za dogodke z različnimi učinki
- prikaži ponavljanja, kjer dogodek vpliva na več pojavitev iste entitete
- dodaj dvosmerne puščice za razmerja učinkov ena-proti-ena
- zlij ponovljive učinke (isti dogodek različno vpliva več množic iz iste entitete)
- dodaj entitete, ki so potrebne zaradi dodatnega povpraševanja med dogodkom
- dodaj vhodne podatke (sprožilec)

## SSADM: Specifikacija zahtev



**Izdelava diagramov EAP (Enquiry Access Paths)**

- prikaz navigacije po podatkovnem modelu pri poizvedovanjih
- formalno preverjamo, če LDS podpira vsa poizvedovanja
- za vsakega izdelamo svoj EAP na osnovi Seznama zahtev in Definicij funkcij
- notacija podobna ECD (Jacksonovi diagrami)
- doslej to lasnost LDS preverjali neformalno v korakih 140 oziroma 320
- EAP nadalje uporabimo pri definiciji procesiranja povpraševanj (korak 530)

**Notacija:**

- elementi so entitete, do katerih dostopamo med povpraševanjem
- enosmerna puščica za opis razmerja dostopanih entitet ena proti ena, smer predstavlja smer dostopa
- sprožilec povpraševanja - enako kot pri ECD, zraven vhodni podatki - ključ
- ponavljanje ali izbira pri dostopih - Jacksonova notacija





## SSADM: Specifikacija zahtev



**8 korakov razvoja EAP:**

- poimenuj povpraševanje
- določi sprožilec povpraševanja
- določi entitete, do katerih dostopamo med izvajanjem povpraševanja
- nariši logično podatkovno strukturo povpraševanja
- dodaj strukture iz Jacksonovih diagramov
- razmerja ena-proti-ena med elementi in entitetami
- prikaz vstopne točke (sprožilca) in podatkov
- preveri povpraševanje

## SSADM: Specifikacija zahtev



**Potrditev sistemskih ciljev (Confirm System Objectives) - korak 370**

**Cilj:** preveriti, če so bile vse nefunkcionalne zahteve upoštevane

**Preverjamo tri izdelke:**

- seznam zahtev
- definicije funkcij
- logični podatkovni model

**Združitev rezultatov specifikacije zahtev (Assemble Requirements Specification) - korak 380**

**V tem koraku:**

- združimo rezultate v zaključni dokument z imenom specifikacija zahtev,
- izvedemo kontrolo konsistentnosti ter
- formalno preverimo kakovost posameznih delov (quality assurance review)



## SSADM: Specifikacija logičnega sistema



**MODUL: Specifikacija logičnega sistema (Logical system specification)**

**Dve fazi:**

**Faza 4: Tehnične opcije sistema (Technical System Options):**

- določitev več možnih tehničnih arhitektur sistema na osnovi Specifikacije zahtev, izbira najprimernejše arhitekture

**Faza 5: Logično načrtovanje (Logical Design):**

- od (tehnične) implementacije neodvisna specifikacija z vidika procesiranja dialogov, ažuriranj in povpraševanj

Fazi izvajata dve ločeni ekipi (eno strokovnjaki za računalniško tehnologijo, drugo sistemski analitiki)

Fazi se običajno izvajata vzporedno, pri tem je pomembna komunikacija

**Skrajni možnosti:** projekt zaradi prezahtevne tehnične realizacije odpovemo ali ohranimo obstoječo tehnično rešitev

## SSADM: Tehnične opcije sistema



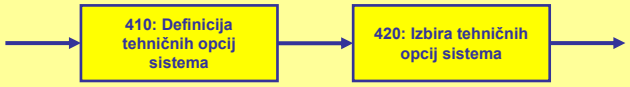
**Faza 4: Tehnične opcije sistema (Technical System Options)**

**Pogoj:** faza specifikacije zahtev mora biti (vsaj v glavnini) zaključena

**Cilj:** razviti več možnih tehničnih arhitektur sistema upoštevaje različne stroške, pridobitve in vplive na organizacijo

Podobnost s fazama Izvedljivosti in Poslovne opcije sistema

**Dva koraka :**




**Izvedba zahteva:**

- izkušene tehnične strokovnjake širokega obsega znanj (za opredelitev strojne in programske opreme, PB, razvojnih okolij, komunikacij)
- vodstveni (poslovni) del ekipe projekta in uporabnikov (za izvedbo analize tveganja, učinka, stroškov, končno odločitev)



## SSADM: Tehnične opcije sistema



**Definicija tehničnih opcij sistema (Define Technical System Options) - korak 410**

Korak je sestavljen iz več zaporednih aktivnosti:

- Identifikacija omejitev (Identify constains)
- Razvoj skic opcij tehnične arhitekture sistema (Develop outline options)
- Podroben opis (izbranih) opcij (Refine each option)
- Ocena performans (izbranih) opcij (Assess performance of each option)

## SSADM: Tehnične opcije sistema



**Izbira tehničnih opcij sistema (Select Technical System Options) - korak 420**

Korak vsebuje naslednje zaporedje aktivnosti:

- Predstavitev in izbira opcije (Present and Select Options): popolne opise opcij predstavimo skupini za odločanje, ki izbere eno izmed opcij
- Dokumentiranje izbrane opcije (Document selected Option): podroben opis opcije v Opisu tehničnega okolja
- Zasnova stila uporabniškega vmesnika aplikacije (Produce Application Style Guide): na osnovi izbrane opcije razvijemo stil za uporabniški vmesnik





## SSADM: Logično načrtovanje



**Faza 5: Logično načrtovanje (Logical design)**

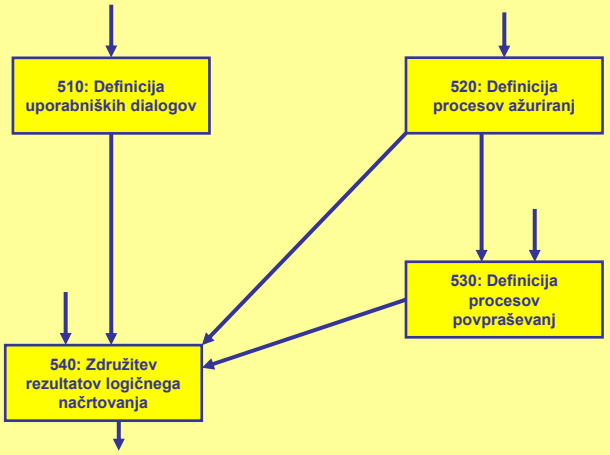
Dodelamo (od opreme neodvisne) specifikacije sistema z vidika komunikacije človek-računalnik in procesiranja:

**Korak 510: Definicija uporabniških dialogov** (Define User Dialogues) - opis strukture in navigacije dialogov, zasnova pomoči

**Korak 520: Definicija procesov ažuriranj** (Define Update Processes) - razvoj modelov procesov ažuriranj, ki podrobneje opisujejo ažuriranja (stanja, operacije)

**Korak 530: Definicija procesov povpraševanj** (Define Enquiry Processes) - podobno kot 520 - razvoj modelov procesov povpraševanj (pogoji, stanja, operacije)

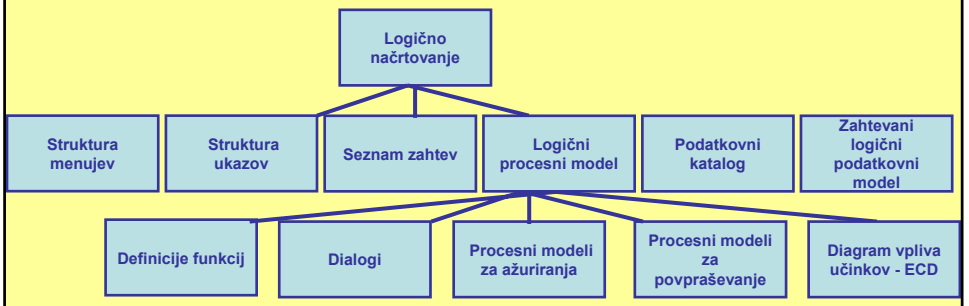
**Korak 540: Združitev rezultatov logičnega načrtovanja** (Assemble Logical Design) - preverjanje ujemanja in popolnosti rezultatov, združitev ter objava le-teh








## Rezultati logičnega načrtovanja



**MODUL: Fizično načrtovanje**

**Faza 6: Fizično načrtovanje (Physical Design)**

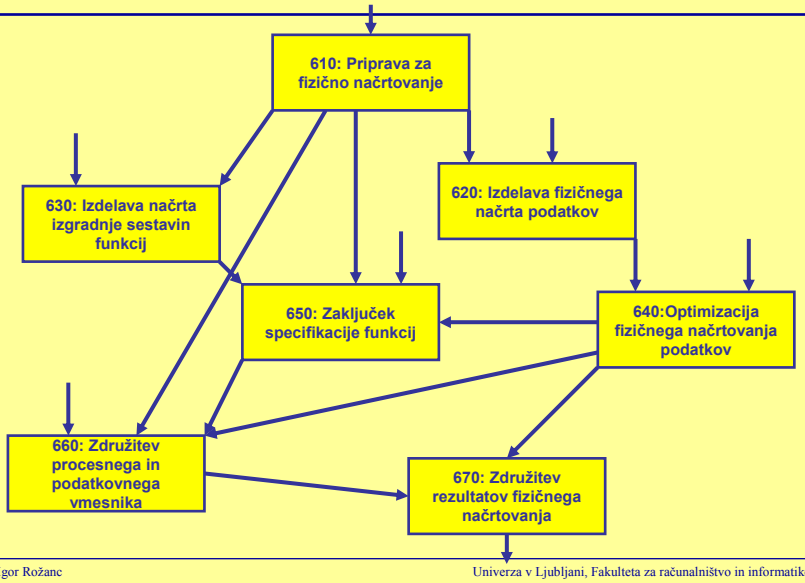
Nadaljevanje razvoja sistema na stopnji, ki je odvisna od izbrane tehnologije (strojne in programske opreme)

**Tri cilji:**

- Zagotoviti, da so narejene vse sistemske komponente in načrti, ki omogočajo izdelavo in vgraditev sistema
- Zagotoviti, da so standardi za izdelavo in vgraditev znani in upoštevani med razvojem sistema
- Izdelati primeren načrt sistema, ki bo omogočal doseganje zahtevanih performans







## SSADM: Fizično načrtovanje



**Sestava faze:**

**Korak 610: Priprava za fizično načrtovanje** (Prepare for Physical Design) - definicija standardov in pristopa pri fizičnem načrtovanju

**Korak 620: Izdelava fizičnega načrta podatkov** (Create Physical Data design) - zahtevani LDS pretvorimo v podatkovni načrt izbrane PB

**Korak 630: Izdelava načrta izgradnje sestavin funkcij** (Create Function Component Implementation Map) - razvoj načrta fizične implementacije funkcij

**Korak 640: Optimizacija fizičnega načrtovanja podatkov** (Optimize Physical data Design) - izboljšava podatkovnega načrta, da doseže zahtevane performanse

**Korak 650: Zaključek specifikacije funkcij** (Complete Function Specification) - dopolnimo specifikacijo funkcij z opisi funkcij, ki jih ne moremo opisati na proceduralen način





## SSADM: Fizično načrtovanje



**Sestava faze (nadaljevanje):**

**Korak 660: Združitev procesnega in podatkovnega vmesnika** (Consolidate Process Data Interface) - definiranje povezave med fizičnim podatkovnim načrtom in procesiranjem podatkov

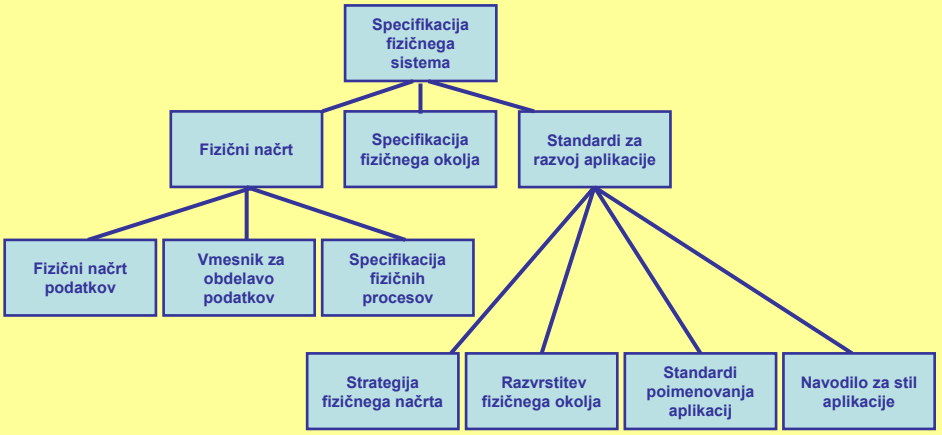
**Korak 670: Združitev rezultatov fizičnega načrtovanja** (Assemble Physical Design) - preverjanje ujemanja in popolnosti rezultatov, združitev vseh rezultatov ter objava le-teh

**Faza fizičnega načrtovanja je zadnja faza SSADM !**

## SSADM: Fizično načrtovanje



### Rezultati faze fizičnega načrtovanja






## SSADM: Zaključek



## SSADM: Literatura




- Mike Goodland, Caroline Slater: SSADM Version 4, A Practical Approach, McGraw-Hill, 1995.
- SSADM Reference Manual, 1986 (Version 3), 1990 (Version 4).
- Downs, Clare, Coe: SSADM, Application and Context (2nd Edition), Prentice-Hall, 1991.
- Philip L. Weaver: Practical SSADM V4, Pitman Publishing, 1993.
- Malcom Eva: SSADM Version 4: A User's Guide, (2nd Ed), McGraw-Hill, 1994.
- Internet ...