
STRUKTURNA SISTEMSKA ANALIZA IN NAČRTOVANJE

(SSADM - STRUCTURED SYSTEMS ANALYSIS AND DESIGN METHOD)

SSADM – cilji in koncepti

- strukturiran razvoj programske opreme
- metodologija
- notacija (pravila zapisovanja)
- integracija analize in načrtovanja
- cilj: nadzor kakovosti
- zasnova (izhodišče): tok informacij (podatkov)
- v 80-ih in 90-ih letih pogosto uporabljana metodologija, danes jo postopoma nadomešča poenotena metodologija (unified process - UML)
- veliko inačic

Glavni koraki

1. pripravi in analiziraj **Diagram Toka Podatkov**
Data Flow Diagram - DFD
2. iz diagrama toka podatkov izpelji
Strukturni graf
Structure Chart - SC

Izhod (output) faze analize

1. Diagram Toka Podatkov (Data Flow Diagram)
2. Podatkovni Slovar (Data Dictionary)
3. Mini Specifikacije (Process Specifications)

Diagram toka podatkov

- Celotni IS je sestavljen iz:
 1. procesov,
 2. poslovne logike in
 3. podatkovnega modela.
- Pri DFD se osredotočimo na:
 1. procese in
 2. tok podatkov med njimi (kako podatki pridejo v sistem, kam gredo, kje se obdelujejo, kdo sprejema odločitve...)

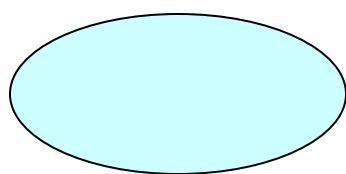
Logični / fizični nivo modela

- Logični nivo
 - opisuje procese, ne da bi pri tem povedal kako so realizirani - izpeljani.
- Fizični nivo
 - vključuje informacije o tem, kako so procesi implementirani

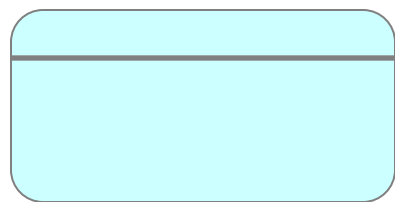
Notacijski elementi DFD

De Marco, Yourdon

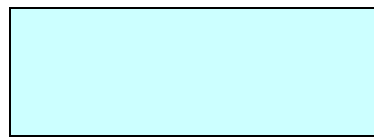
Gane, Sarson



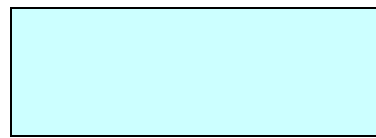
proces



zbirka podatkov



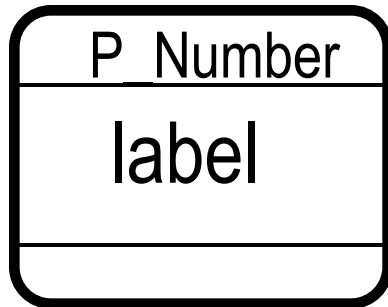
izvor / ponor
(terminator)



tok podatkov

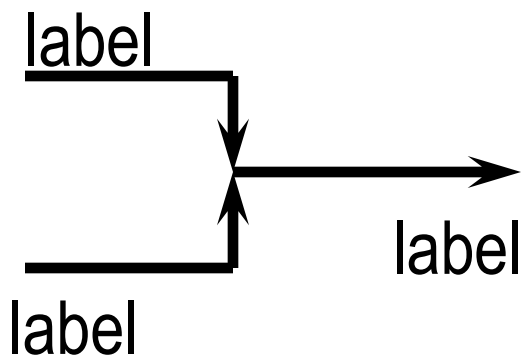
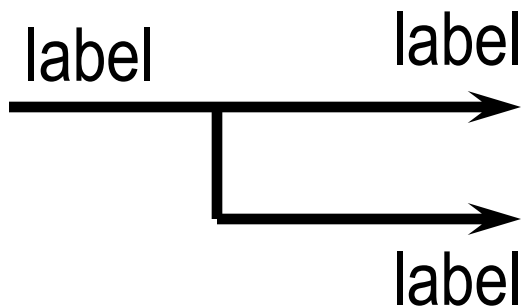
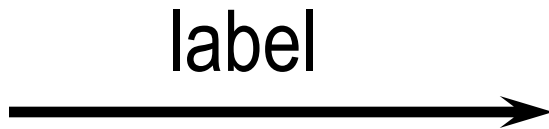


Procesi



- proces = aktivnost
- predstavlja transformacijo podatkov
- na vhodu sprejme podatkovni tok
- na izhodu proizvaja podatkovni tok
- lahko ga naprej razgradimo (razčlenimo/ izvedemo dekompozicijo)
- labela: glagol
- povezan je s specifikacijo procesa

Tok podatkov



- predstavlja prehod informacij
- povezovalni element
- prenos podatkov
- ne predstavlja nadzor pretoka
- je usmerjen
- lahko se deli ali združi
- labela: samostalnik
- opisan je v podatkovnem slovarju

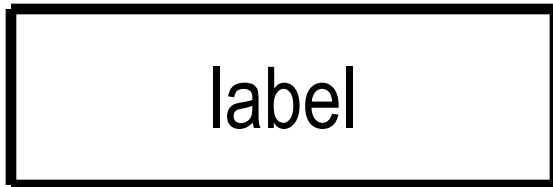
Zbirka podatkov

label

D#	label
----	-------

- trajna shramba za podatke
- predstavlja bazo podatkov ali datoteke
- pasivnost – v bazi se ne izvajajo aktivnosti
- obdelana s strani procesov
- labela: samostalnik
- opisana je v podatkovnem slovarju in/ali z ER diagramom

Terminator



- izvor / ponor za zunanje tokove podatkov
- zagotavlja povezavo z vsebino sistema
- pasivnost: razen sprejemanja / pošiljanja podatkov ne izvaja nobene druge aktivnosti
- labela: samostalnik
- opisana je v podatkovnem slovarju

Postopek izdelave DFD

1. izdelava kontekstnega diagrama
2. izdelava diagrama na 'prvem' nivoju
3. funkcionalna dekompozicija posameznih procesov do elementarnih procesov
4. validacija diagrama z uporabnikom

Značilnosti kontekstnega diagrama

- Ima le en proces → cel poslovni sistem se nahaja znotraj edinega procesa z oznako 0
- ni zbirk podatkov → vse so skrite znotraj edinega procesa
- izvori / ponori so (določajo) meje sistema (zato se imenujejo tudi terminatorji)

Primer: Kontekstni diagram knjigarne

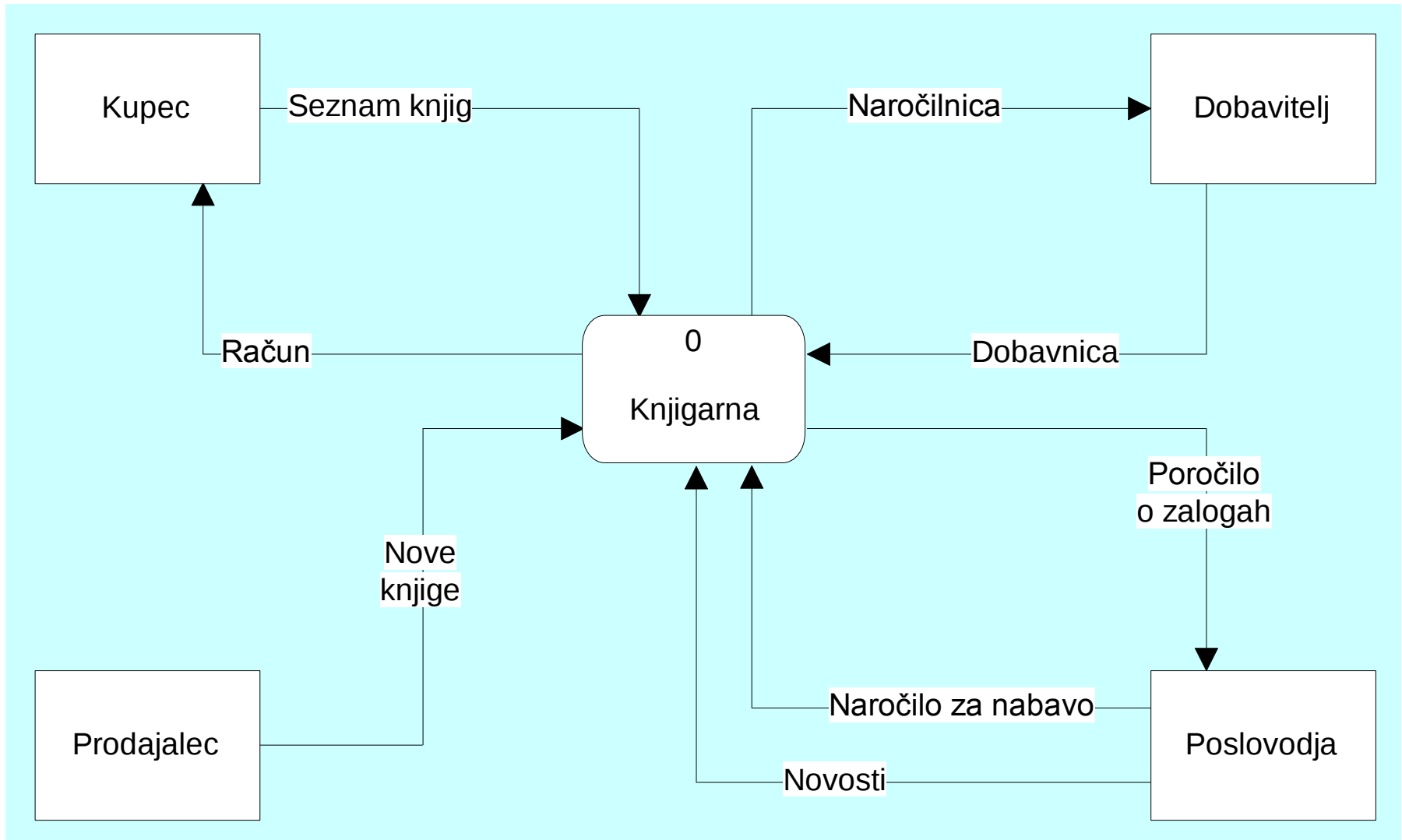


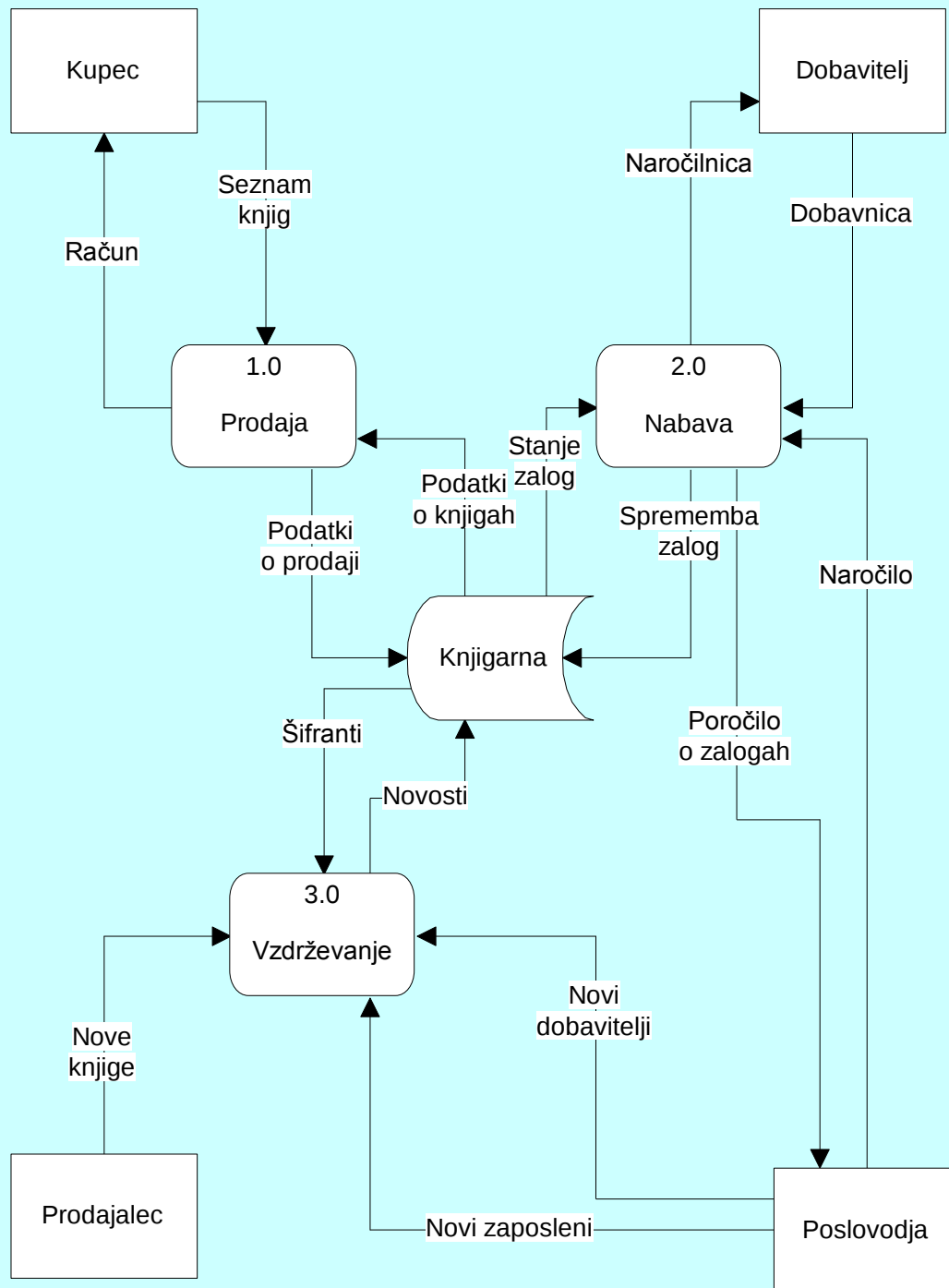
Diagram na 'prvem' nivoju

Diagram prikazuje podprocese (funkcije) osnovnega procesa.

Primer: *iz opisa so razvidne tri funkcije (procesi):*

- *prodaja,*
- *nabava in*
- *vzdrževanje.*

Primer:
DFD
diagram
na 1.
nivoju za
knjigarno



Funkcionalna dekompozicija

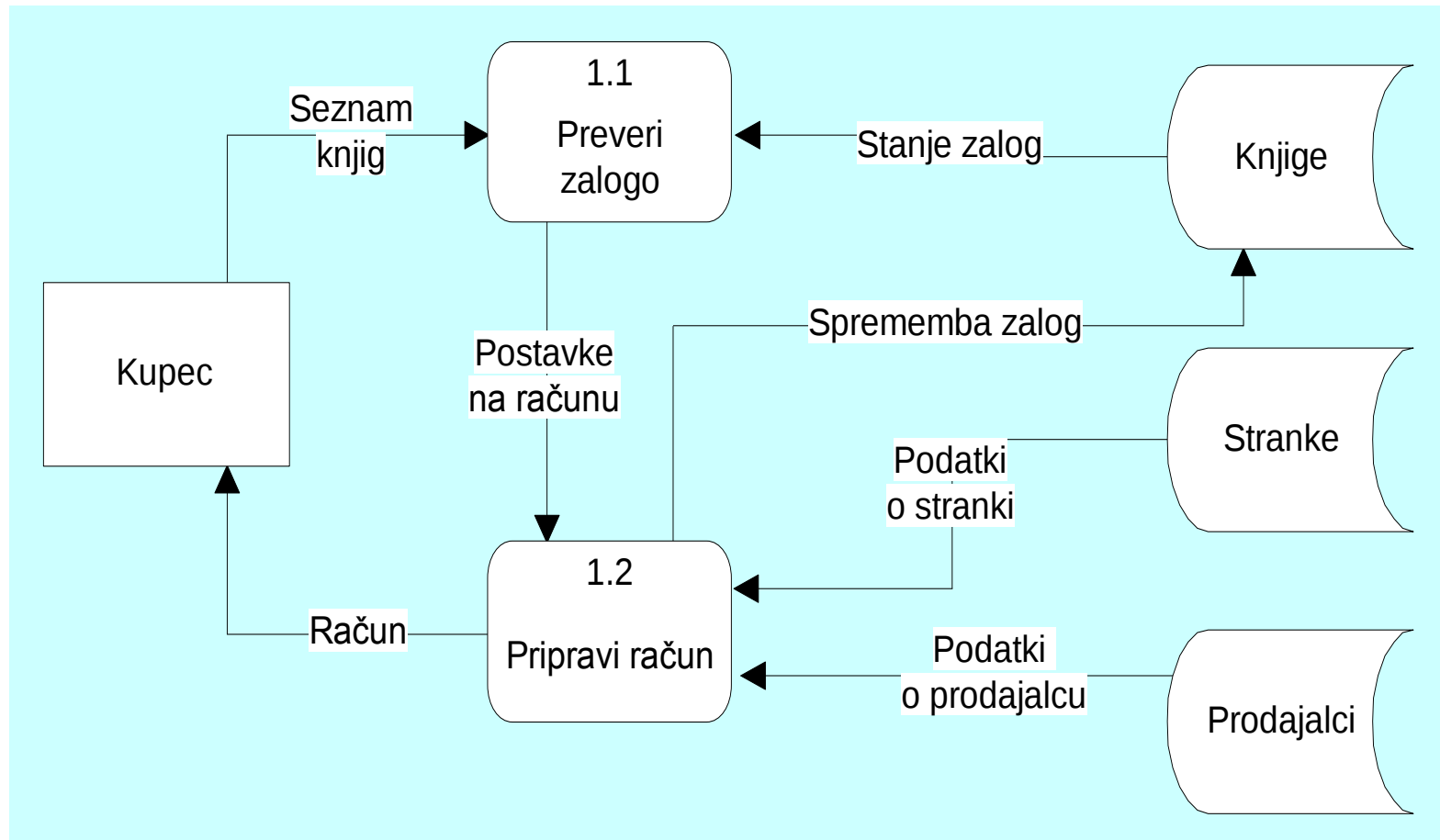
Posamezne procese proučimo in napravimo diagrame toka podatkov na nižjih nivojih.

podprocese označimo s številkami – denimo:

1.1.3. pomeni tretji nivo – to je tretja podfunkcija prve podfunkcije prve funkcije

- na nižjih nivojih novih izvorov / ponorov ni!
- pri dekompoziciji pazi na število povezav:
 - proces, definiran na višjem nivoju mora imeti enako število vhodov in izhodov kot je vsota vhodov in izhodov vseh njegovih podprocesov

Primer: dekompozicija procesa 1. na podprocese



Kdaj je dekompozicija končana?

- ko pridemo do **elementarnih procesov** -
 - elementarni procesi predstavljajo en izračun, eno odločitev, eno operacijo nad bazo...
- ko je vsak tok ena transakcija, on-line dialog, poročilo, dokument...
- ko zbirka predstavlja entiteto
- ko je uporabnik zadovoljen ali pa ko vidimo, da lahko izvajamo naslednje faze izgradnje

Konceptualni standardi podatkovnih tokov

1. podatkovni tokovi ponazarjajo potek informacije od enega objekta k drugemu objektu
2. podatkovni tokovi ne morejo spreminjati podatkov
3. podatki ne morejo biti ustvarjeni ali uničeni znotraj podatkovnega toka
4. podatkovni tok je lahko v določenih pogojih prazen
5. podatkovni tokovi ne morejo neposredno prehajati od terminatorja (vir/ponor) k datoteki
6. podatkovni tokovi ne morejo neposredno prehajati od terminatorja do terminatorja

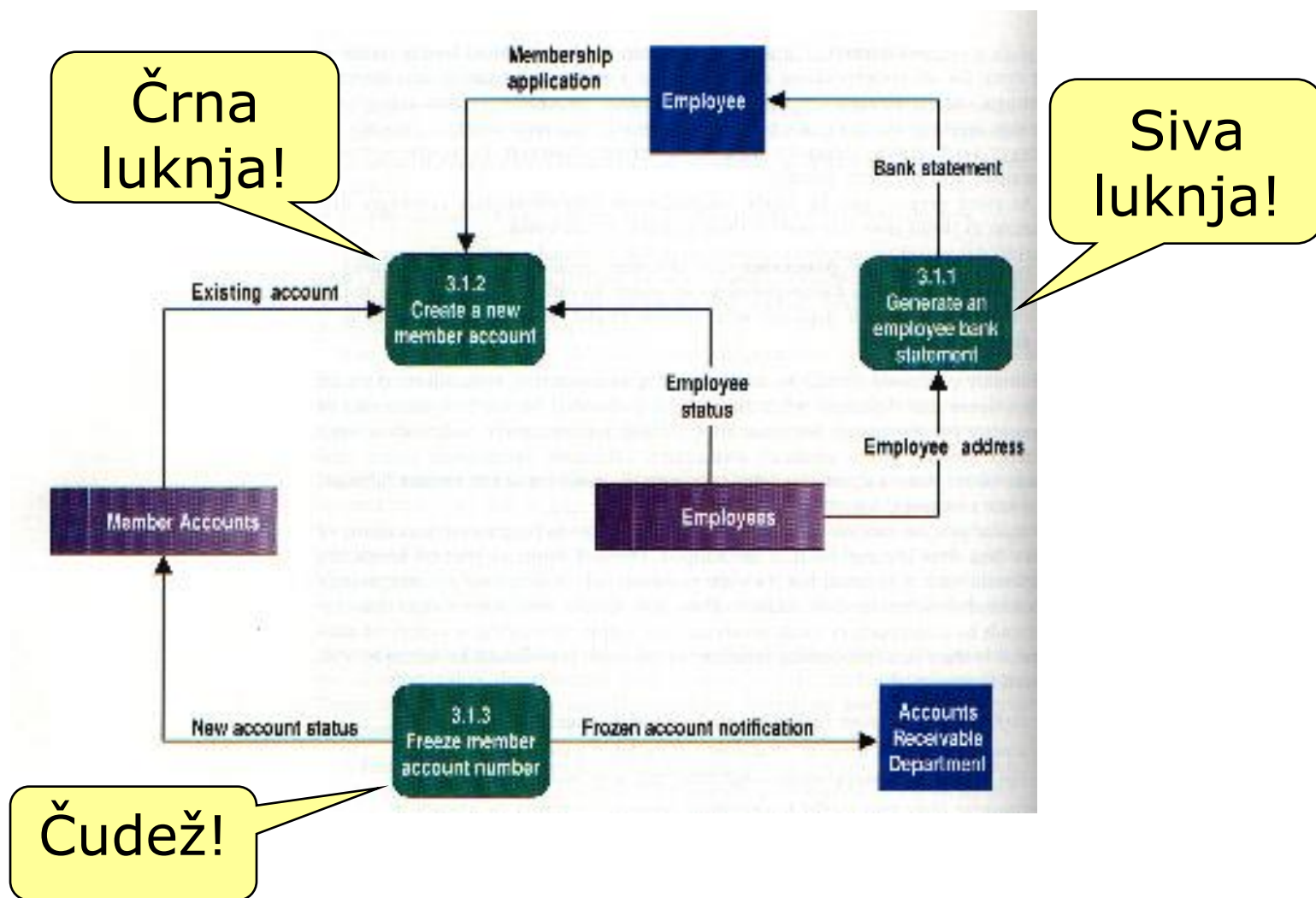
Notacijski standardi podatkovnih tokov in zbirk podatkov

- podatkovni tok je predstavljen z usmerjeno črto, opremljeno z labelo
- vsak podatkovni tok mora biti označen z labelo.
 - Izjema: podatkovni tok, usmerjen proč od zbirke ali v zbirko, ni treba poimenovati, če predstavlja zbirko v celoti.
- labela je samostalnik, izogibamo se generičnim računalniškim imenom (DATA, INPUT, ...)
- vsi podatkovni tokovi morajo nastopati v podatkovnem slovarju
- zbirka mora imeti vsaj en vhodni in vsaj en izhodni tok
- zbirka se lahko pojavi na katerikoli plasti (nivoju) DFD

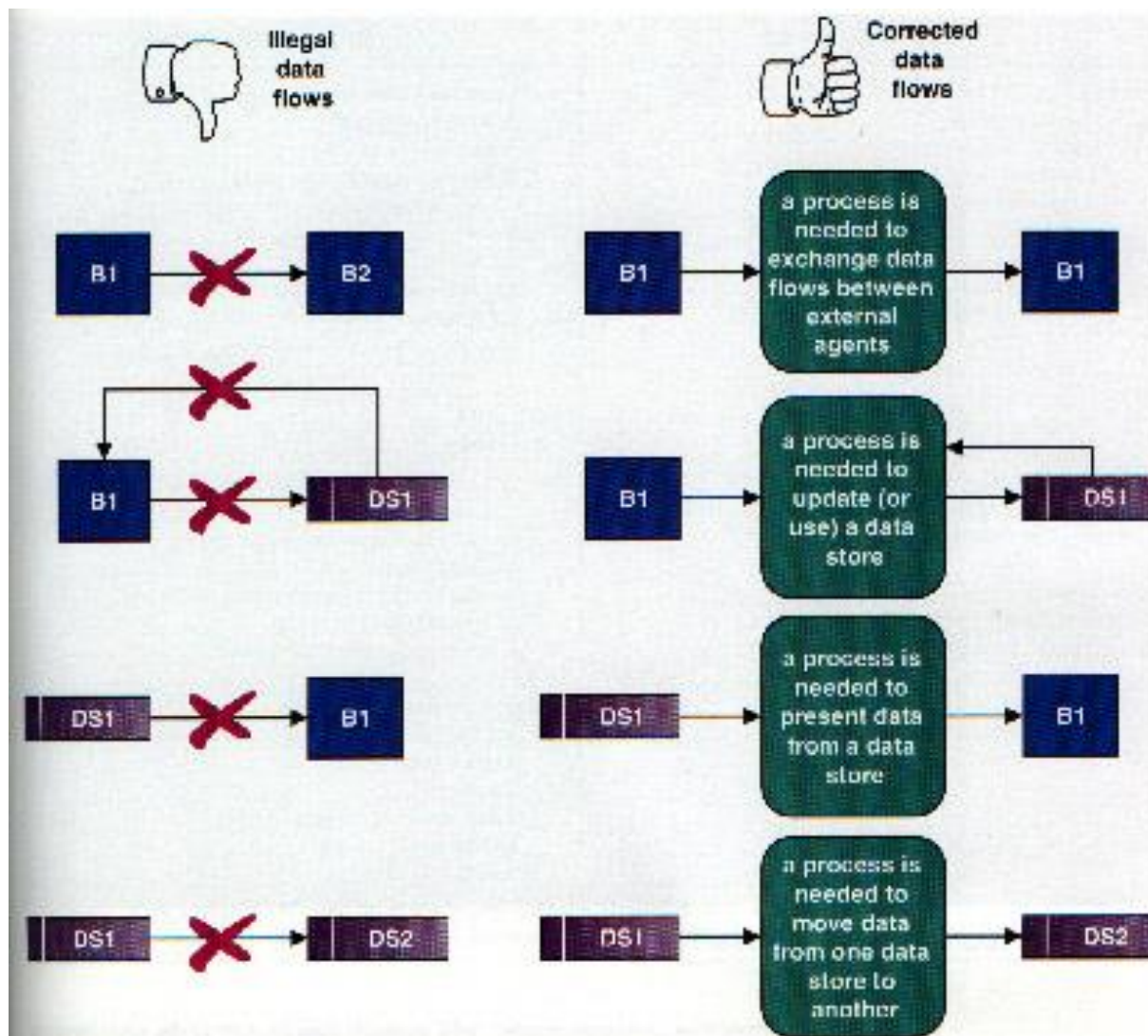
Nekaj splošnih navodil za DFD

- dekompozicija = razčlenitev procesa na elementarne sestavine
- vmesniki med procesi morajo tudi na nižjih nivojih ostati konsistentni (skladni)
- pri procesih na nižjih nivojih lahko dodamo nove tokove podatkov in nove zbirke podatkov
- nivo 0 uporabljamo kot 'abstraktni' nivo
- terminatorji niso procesi (niso aktivni)
- DFD nima časovne komponente

DFD – nepravilni primeri

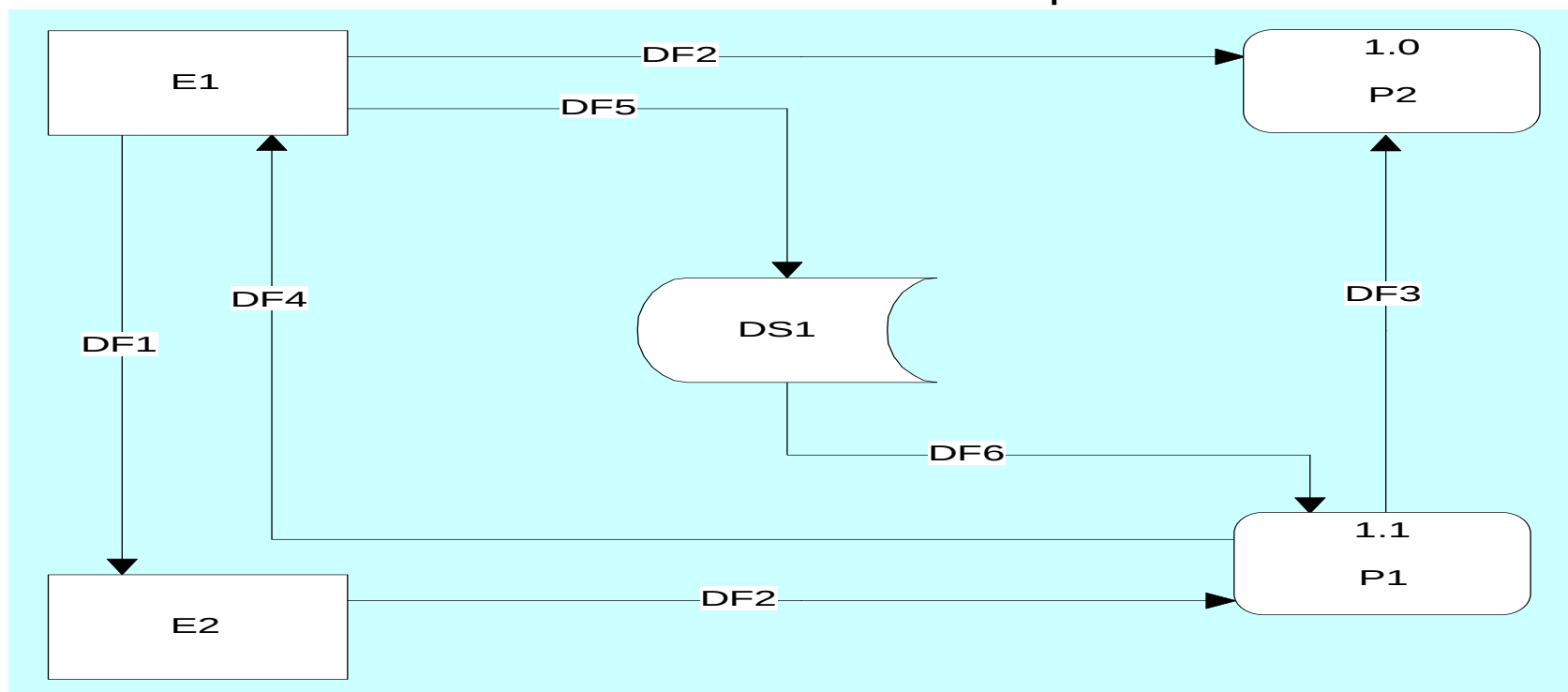


Neveljavni podatkovni tokovi

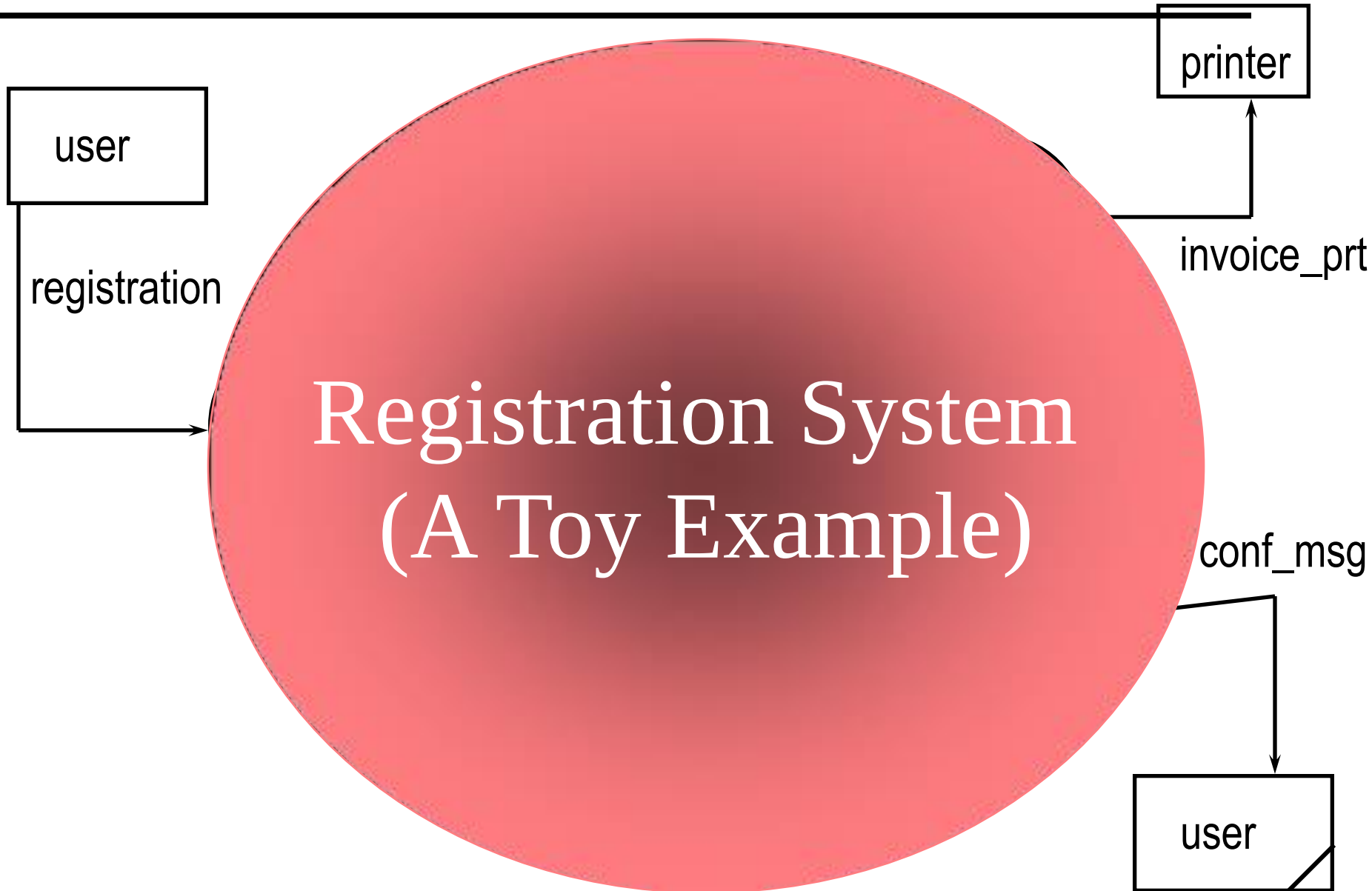


Poiščimo napake:

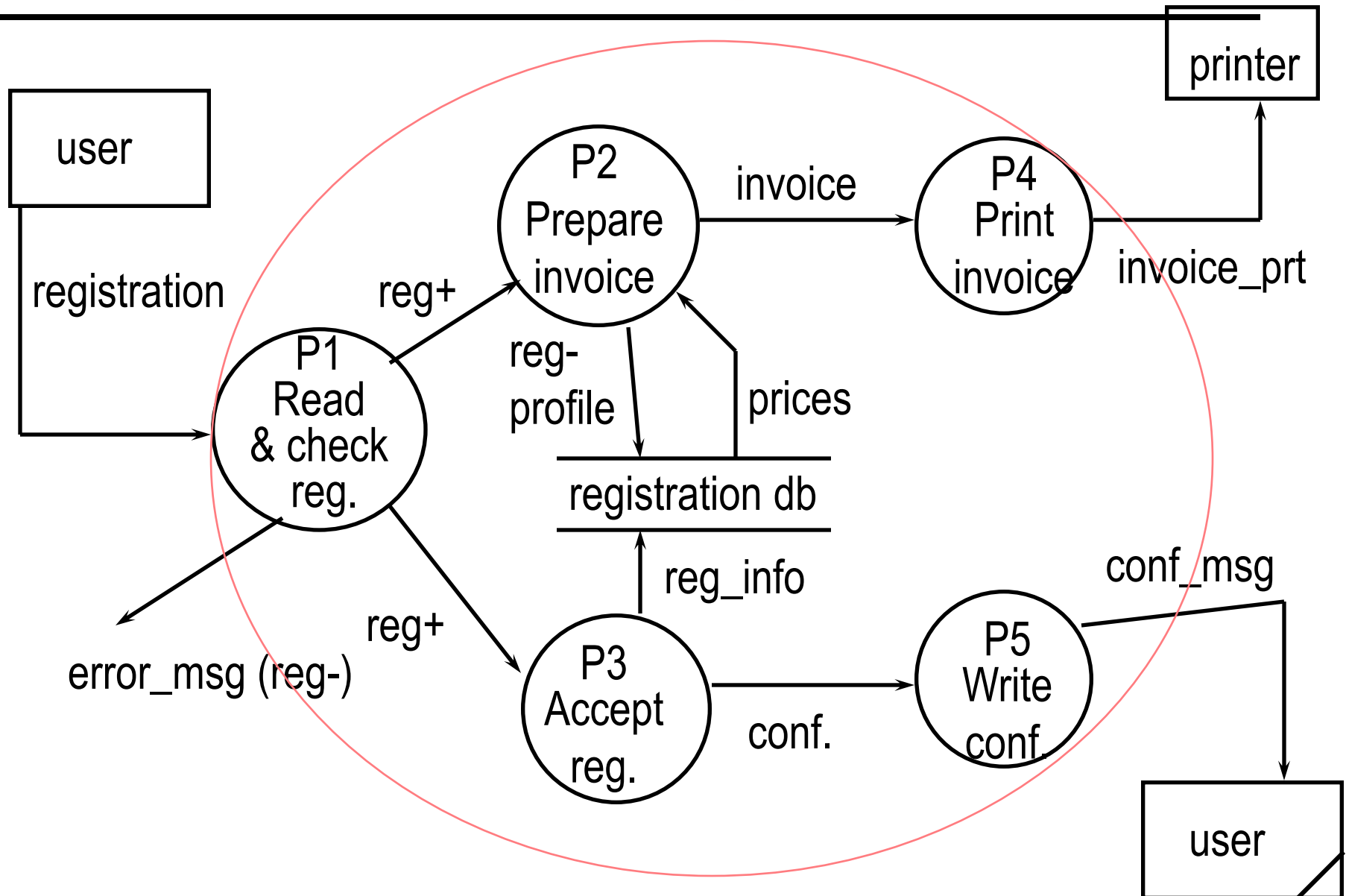
1. All processes must have at least one data flow in and one data flow out.
2. All processes should modify the incoming data, producing new forms of outgoing data.
3. Each data store must be involved with at least one data flow.
4. Each external entity must be involved with at least one data flow.
5. A data flow must be attached to at least one process.



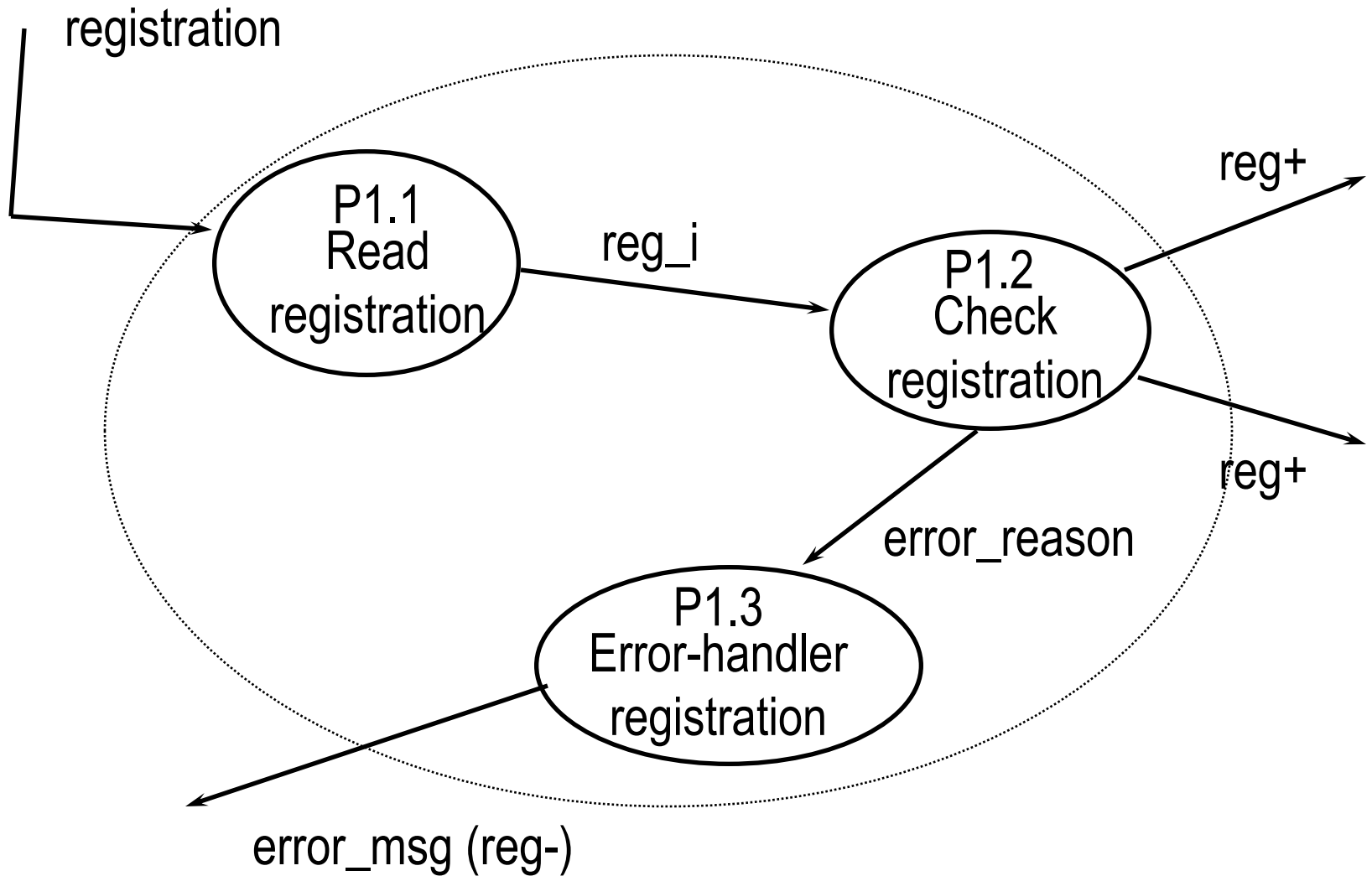
DFD 0: Registration System (*Context Level*)



DFD L1: Registration System



DFD L2: P1- Read and Check Registration



Podatkovni slovar

- specificira (opredeli) podatkovne elemente DFD:
 - tokove podatkov
 - enostavne zbirke podatkov
- skladnost z DFD
 - specifikacije morajo biti skladne z diagramom
- popolnost
 - glede na zahteve DFD mora biti popoln
- sestavljajo ga specifikacije podatkovnih elementov
- vsi podatkovni tokovi in zbirke podatkov iz DFD so razčlenjeni do podatkovnih elementov oz. prvinskih podatkov (DE - "data element")

Formalna notacija za podatkovni slovar

- **pravila** - opredeljujejo opis, strukturo ali zamenjavo
= izenačitev - kompozicija
podatkovni element = podatkovna struktura
oseba = ime_osebe + naslov
ime_osebe = priimek + osebno_ime
- **meta-simboli** - opisujejo strukturo:
 - + logični in (zaporedje)
 - [...|...|...] logični ekskluzivni ali
 - m{ ... }n iteracija ($\geq$ ena ponovitev)
 - (...) opcija
 - * komentar

1. primer podatkovnega slovarja

naročeno_blogo=šifra_blaga + naročena_količina +
prevoz + (opombe)

prevoz = [letalo | ladja | vozilo]

letalo = prevoznik + številka_leta + destinacija + datum +
(spremljevalec)

2. primer podatkovnega slovarja

* registracija osebe za tečaj-e

registracija = oseba + podjetje + (1{tečaj}4) + datum +
(opomba)

oseba = ime + priimek + (naziv)

podjetje = ime_podjetja + ime_oddelka + naslov

naslov = ulica + hišna_št + poštna_št + kraj + (država)

tečaj = ["word" | "excel" | "powerpoint" | "internet"]

=> alternativna obravnava tečajev?

- alternativa:

registracija = oseba + podjetje + tečaji + datum +
(opomba)

oseba = ime + priimek + (naziv)

podjetje = ime_podjetja + ime_oddelka + naslov

naslov = ulica + hišna_št + poštna_št + kraj + (država)

tečaji = ("word") + ("excel") + ("powerpoint") + ("internet")

Data Dictionary - naloge

- *Napišite podatkovni slovar za voziško dovoljenje*
- *Napišite podatkovni slovar za teden v šolskem urniku*

Mini Specifikacije

- izdelamo za tiste procese iz DTP, ki niso razgrajeni, torej so **atomarni** (prvinski) procesi, imenovani tudi funkcionalne primitive.
- opišemo **transformacijo** vhodnih podatkov v izhodne
- vsak elementarni proces ima svojo MS.
- za MS lahko uporabimo:
 - **strukturirano besedilo**
 - **odločitvene tabele**
 - **odločitvena drevesa**

MS - Strukturirano besedilo

- za formaliziranje zapisa so uporabljeni osnovni proceduralni konstrukti:
 - **zaporedje** / sekvenca (blok)
 - **odločitev** / vejitev (if – then – else)
 - **ponavljanje** / zanka (while, repeat)
- kot operandi nastopajo tisti podatkovni tokovi, ki prihajajo v ali izstopajo iz trenutno aktualnega procesa

Primer strukturiranega besedila

proces *Izdaja_čkov*
številka procesa: *X.X*

begin

if stanje pozitivno **then**

izdaj 20 čkov

else (stanje je negativno)

if dovoljena prekoračitev **then**

izdaj 10 čkov

opozorilo o negativnem stanju

else (prekoračitev ni dovoljena)

opozorilo o negativnem stanju

end

Odločitvena tabela - shema

(ukrep-akcija)

popis pogojev		vrednosti pogojev
. . .		. . .
-----		-----
popis ukrepov		ukrepi pri pogojih
. . .		. . .

- odločitvene tabele lahko kombiniramo s strukturnim besedilom

Odločitvena tabela - shema

Popis pogojev	pogoj1	vr11	vr12	vr13	...	
	pogoj2	vr21	vr22	vr23	...	
	...	...	...	..		
Popis ukrepov	ukrep1	u1kp1	u1kp2	u1kp3	...	
	ukrep2	u2kp1	u2kp2	u2kp3	...	
		...	...	...	...	

- u1kp1 pomeni ukrep 1 pri kombinaciji pogojev 1 (prvi stolpec)

Odločitvena tabela – primer izdaja čekov

Popis pogojev	pozitivno stanje	D	D	N	N
	prekoračitev dovoljena	N	D	D	N
Popis ukrepov	število izdanih čekov	20	20	10	0
	obvestilo o ned. stanju	N	N	D	D

DODATNE PROSOJNICE