
OSNOVE INFORMACIJSKIH SISTEMOV

2. letnik, visokošolski strokovni študij
smer PROGRAMSKA OPREMA



UNIVERZA V LJUBLJANI
Fakulteta za računalništvo in informatiko



infolab.fri.uni-lj.si
InfoLab fakultete za računalništvo in informatiko

Splošne informacije ⁽¹⁾

- Predavatelj:
 - Viš. pred. dr. Rok Rupnik
Rok.rupnik@fri.uni-lj.si
- Asistent
 - mag. Alenka Rožanec
Alenka.rozanec@fri.uni-lj.si



infolab.fri.uni-lj.si
InfoLab fakultete za računalništvo in informatiko

Splošne informacije (2)

Namen predmeta ◊

- Namen predmeta
 - Osnovni namen predmeta je naučiti se:
 - osnovnih pojmov v povezavi z informacijskimi sistemi: kaj je informatika, kaj je splošni sistem, kaj poslovni sistem, kakšni procesi se izvajajo v poslovnem sistemu, kakšna je razlika med podatkom in informacijo, kaj je informacijski sistem, kakšne so njegove značilnosti in kakšna je njegova vloga v poslovnem sistemu.
 - sistematičnih pristopov k razvoju kakovostnih informacijskih sistemov. Skozi vsebino predmeta spoznamo življenjske cikle razvoja informacijskih sistemov, različne metode in tehnike izvajanja posameznih korakov razvoja, strukturni in objektni razvoj, razne metodologije, strateško planiranje ipd.

Splošne informacije (3)

Priporočena literatura ◊

- Priporočena literatura
 - STEVEN ALTER (1999). *Information Systems - A Management Perspective*, third edition, Addison-Wesley.
 - BOOCH, G., J. RUMBAUGH in I. JACOBSON (1999). *The Unified Modelling Language - User Guide*. Addison Wesley Longman, Inc.
 - JACOBSON, I., G. BOOCH in J. RUMBAUGH (1999). *The Unified Software Development Process*. Addison Wesley Longman, Inc.
 - CVI-Ljubljana, FRI-Ljubljana, IPMIT (2004). *EMRIS-Enotna metodologija razvoja IS*, druga izdaja, CVI-Ljubljana.

Vsebina predmeta ⁽¹⁾

1. Predstavitev predmeta
2. Opredelitev osnov informacijskih sistemov
3. Splošno o razvoju informacijskih sistemov
4. Strukturni pristop
5. Objektni pristop
6. Vabljen predavanja

1 - predstavitev predmeta

Vsebina predmeta ⁽²⁾

4. Strukturni pristop
 - Osnovne značilnosti
 - Primer strukturnega pristopa: *IE-Information Engineering*
5. Objektni pristop
 - Osnovne značilnosti
 - Modelirni jezik *UML-Unified Modeling Language*
 - Primer objektnega pristopa: *RUP-Rational Unified Process*
6. Informacijske tehnologije
 - Podatkovna skladišča
 - Portali
 - Elektronsko poslovanje
 - Arhitekture informacijskih sistemov
 - Zagotavljanje varnosti

Poglavje 2

OPREDELITEV OSNOV INFORMACIJSKIH SISTEMOV

Informatika

- Informatika je znanstvena disciplina, ki raziskuje zgradbo, funkcije, zasnovo, organiziranje in delovanje informacijskih sistemov.
- Študija IS je multidisciplinarno področje. Glavne veje, ki se z IS ukvarjajo, so:

- Računalništvo (*Computer Science*)
- Upravljanje (*Management Science*)
- Operacijske raziskave (*Operations Research*)

Tehnični pristop

- Sociologija (*Sociology*)
- Politologija (*Political Science*)
- Psihologija (*Psychology*)

Vedenjski pristop

Računalništvo in informatika

- *Računalništvo (Computer Science)*
 - Računalnik je predmet proučevanja
- *Informatika (Information Science)*
 - Računalnik je sredstvo za dosego cilja

INFORMATIKA = INFORMACIJA + AVTOMATIKA

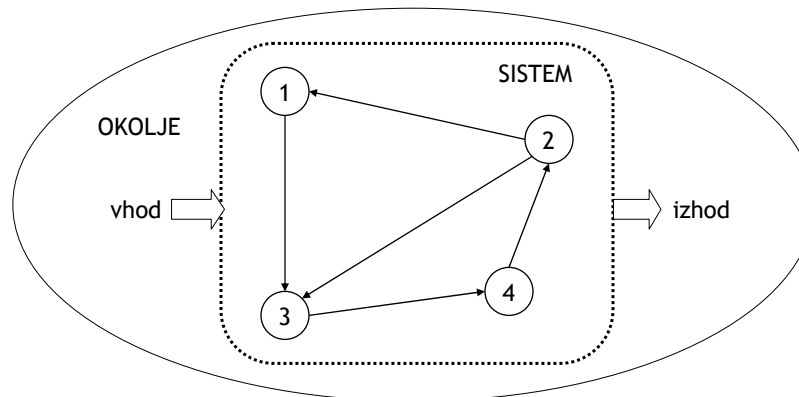
Informacijska družba

- *Informacijska družba* je sinonim za novo nastajajočo družbo, ki ne temelji zgolj na izkoriščanju surovin in energije, temveč kot najpomembnejši vir jemlje informacije in znanje.
- *Informacijska družba* je izrazito storitveno naravnana družba, kjer je uspešnost tako posameznika kot tudi organizacije temelji na sposobnosti pridobiti čimveč različnih podatkov ob pravem času, iz njih hitro izluščiti pravilno informacijo na njeni podlagi učinkovito ukrepati

Splošno o sistemih

- Sistem je celota, ki se sestoji iz več komponent ali podsistemov in množice povezav med njimi.
- S sistemi je mogoče ponazoriti vse človekovo notranje in zunanje okolje.
- Sisteme lahko razdelimo v tri temeljne skupine:
 - **NARAVNI SISTEMI:** so iz naravnih sestavin, delujejo po naravnih zakonitostih, za naravne smotre; uravnavajo se sami
 - **TEHNIČNI SISTEMI:** iz naravnih snovi jih snuje človekov razum, uporabljajo naravne zakonitosti in delujejo za cilje organizacije; krmiljenje je avtomatizirano, samodejno
 - **ORGANIZACIJSKI SISTEMI:** jih snuje človekov razum iz naravnih in tehničnih sistemov, delujejo po načelih in predpisih za smotre in cilje organizacije; krmiljenje je zavestno - izhaja iz človekove volje.
- S sistemi se ukvarja *teorija sistemov*

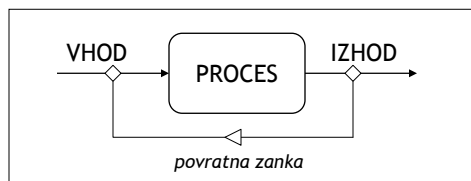
Shema sistema



- Vsaka komponenta je za sistem pomembna - obstoj in funkcija komponente vplivata na obstoj in funkcijo celotnega sistema.
- Nobena komponenta ni *izolirana*.
- Sistem s svojo funkcijo vpliva na funkcijo komponente.

Shema sistema (2)

- OKOLJE sistema
 - Okolje sistema je množica komponent, ki so v interakciji s sistemom, vendar niso del sistema.
- VHOD in IZHOD sistema
 - Sistem deluje z določenim namenom oz. za doseg določenega cilja. K cilju strmi z izvajanjem procesa, s pomočjo katerega pretvarja *vhod* v *izhod*.



Kaj nas pri obravnavi sistema zanima?

- Namen sistema
 - namen sistema opredeljuje razlog za obstoj sistema
- Meje sistema
 - meje sistema določajo, kaj je znotraj in kaj zunaj sistema
- Okolje sistema
 - okolje sistema je vse, kar je izven meja sistema in s sistemom sodeluje, oziroma je za sistem pomembno.
- Vhod in izhod sistema
 - vhod in izhod sistema tvorijo fizični objekti (material, blago ipd) in podatki, ki bodisi prihajajo v sistem (vhod) ali gredo iz sistema v okolje (izhod).

Pomembne lastnosti sistema

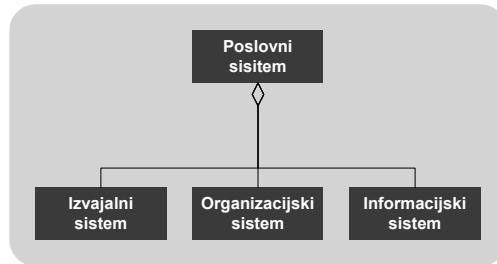
- Entropija
 - Entropija sistema je mera *nereda* v sistemu. Je funkcija verjetnosti stanja sistema.
- Dinamično ravnovesje
 - Sposobnost sistema, da se kljub različnim vplivom in motnjam, vselej znajde v nekem stanju relativne stabilnosti, imenujemo dinamično ravnovesje.
- Prilagodljivost
 - Sposobnost sistema, da spreminja sebe ali povzroča spreminjanje okolja, v primeru, ko je lastno obnašanje sistema ali okolja njemu škodljivo.
- Povratna zveza
 - Povratna zveza je mehanizem, ki omogoča oz. ohranja dinamično ravnovesje v sistemu. Je temeljni mehanizem kontrole delovanja dinamičnega sistema

Poslovni sistem

- Poslovni sistem (*business system, work system*) opredelimo kot sistem, v katerem sodelujejo ljudje kot izvajalci poslovnih procesov in pri tem uporabljajo informacije, tehnologijo in druga sredstva za produkcijo dobrin (izdelkov in storitev) za notranje ali zunanje stranke.
- Temeljni proces poslovnega sistema je *reprodukcijski proces*, temeljni cilj pa *ekonomski*.

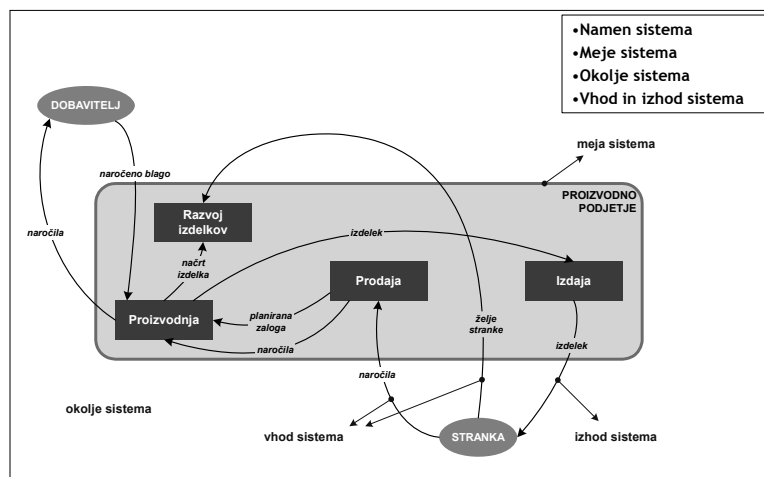
Sestavine poslovnega sistema

- Poslovni sistem je razčlenjen na izvajalni, organizacijski in informacijski podsistem, od katerih vsak zase spet nastopa kot delni sistem.



Proizvodno podjetje kot PS

Proizvodno podjetje kot poslovni sistem s štirimi podsistemi

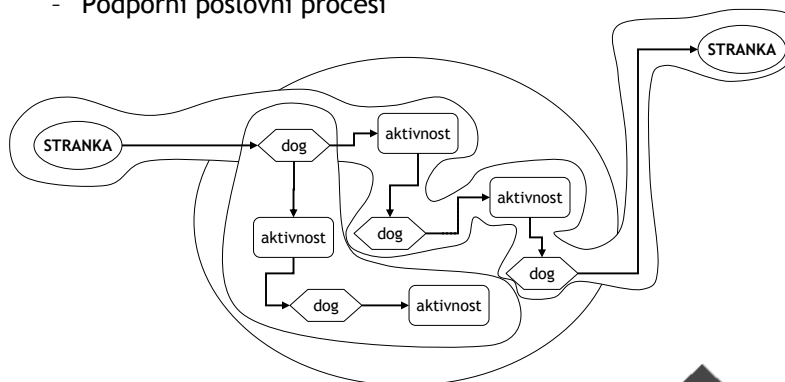


Poslovni procesi v poslovnem sistemu

- Poslovni proces je
 - povezana skupina korakov oziroma aktivnosti, ki se izvajajo v poslovnem sistemu in posredno ali neposredno vplivajo na dodano vrednost pri uresničevanju skupnega cilja poslovnega sistema.
- Podproces in aktivnosti
- Obseg in dodana vrednost poslovnega procesa
- Prenovitev poslovnih procesov

Ključni in podporni poslovni procesi

- Poslovni procesi:
 - Ključni poslovni procesi
 - Podporni poslovni procesi



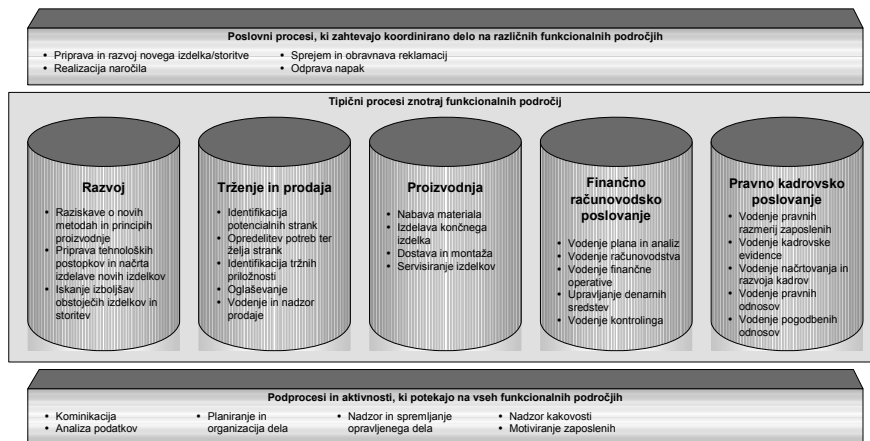
Poslovni procesi v poslovnem sistemu - funkcionalna vs. procesna orientiranost

- Poslovni sistemi se tradicionalno organizirajo okrog funkcionalnih področij (prodaja, nabava, proizvodnja,...)
- Funkcionalno usmerjena organizacija temelji na profesionalizmu in ekspertizi.
 - avtonomnost
 - vsako področje svoje kadre
 - strokovnjaki za posamezna področja
 - usmeritev navznoter
 - vodenje s pomočjo funkcionalnih silosov
 - negativen vpliv na procese, ki potekajo čez več silosov

Poslovni procesi v poslovnem sistemu - funkcionalna vs. procesna orientiranost

- Slabosti funkcionalne orientiranosti silijo podjetja v procesno orientiranost:
 - izbira najpomembnejših poslovnih procesov
 - skrb za optimalno izvajanje poslovnih procesov
 - poslovni procesi imajo lahko svojega skrbnika oziroma vodjo

Poslovni procesi in funkcionalna področja



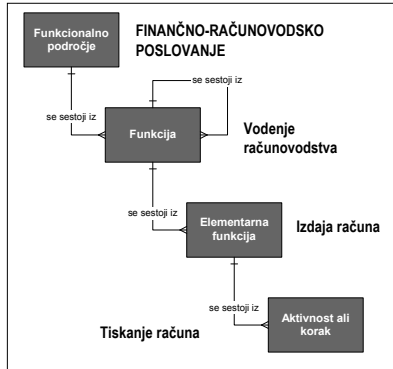
Potek poslovnega procesa

- Potek poslovnega procesa lahko prikažemo na različnih ravneh funkcionalne razgradnje:
 - Na ravni elementarnih funkcij
 - Na ravni aktivnosti
 - Na ravni funkcij
- Odločitev o primerni predstavitvi zavisi od nivoja podrobnosti, ki jih želimo v predstavitvi poteka zajeti.

Pogled na organizacijo kot sistem

Poslovni procesi v poslovnem sistemu - opis poteka poslovnega procesa

- Razdelitev funkcij po funkcionalnih področjih prikažemo s funkcionalno dekompozicijo ali razgradnjo.



Funkcije se združujejo v funkcionalna področja. Navadno so to glavna področja dejavnosti nekega podjetja.

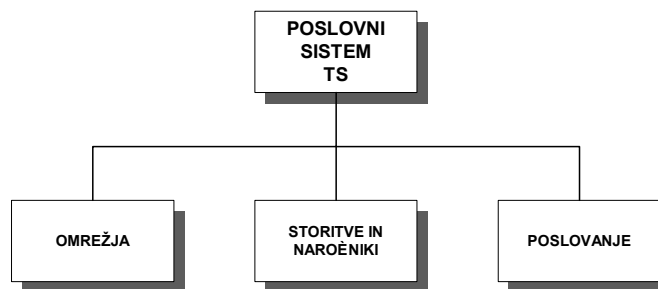
Vsaka elementarna funkcija ima točno določen vhod in izhod.

Pri funkciji ni mogoče opredeliti začetka in konca izvajanja in tudi ne določiti posameznih primerov izvedba funkcije.

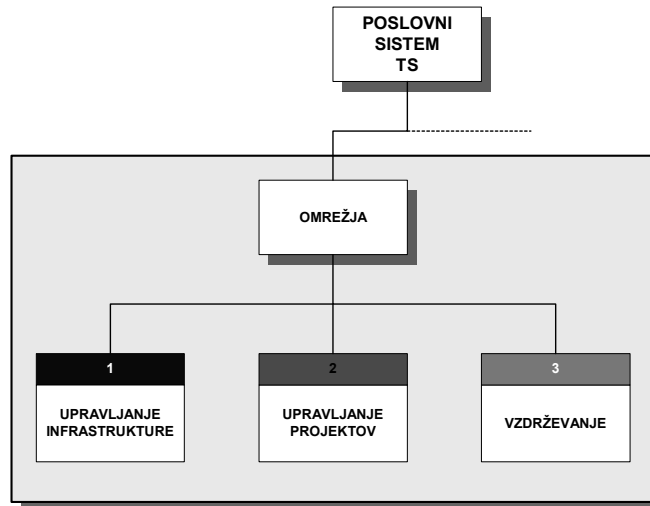
Elementarne funkcije so na najnižjem nivoju sestavljene iz aktivnosti oziroma korakov.

Primer

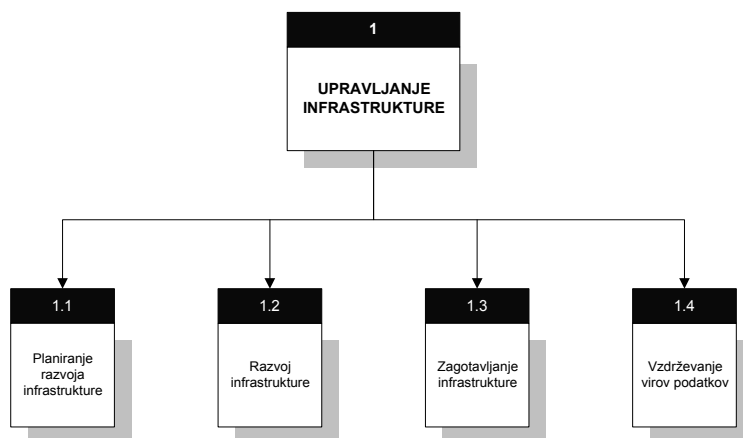
Funkcionalna razgradnja poslovnega sistema TS



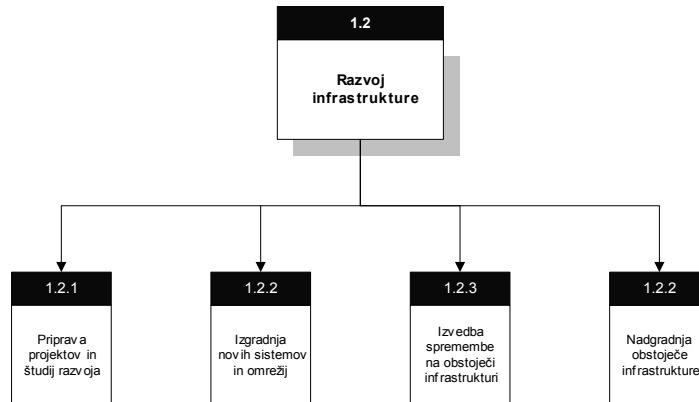
Primer Funkcionalna razgradnja poslovnega sistema TS



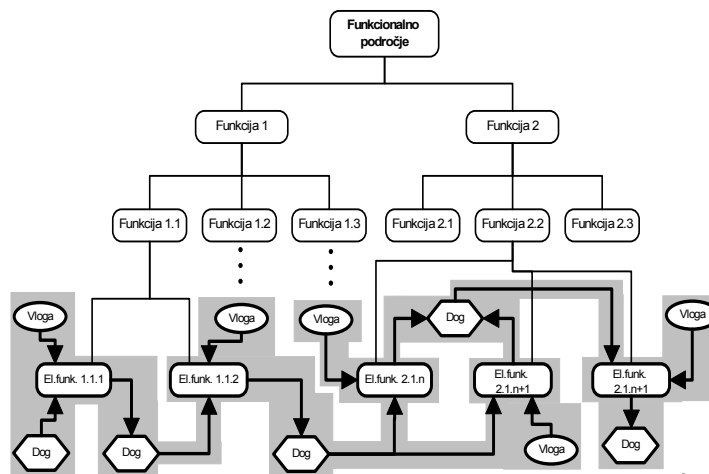
Primer Funkcionalna razgradnja poslovnega sistema TS



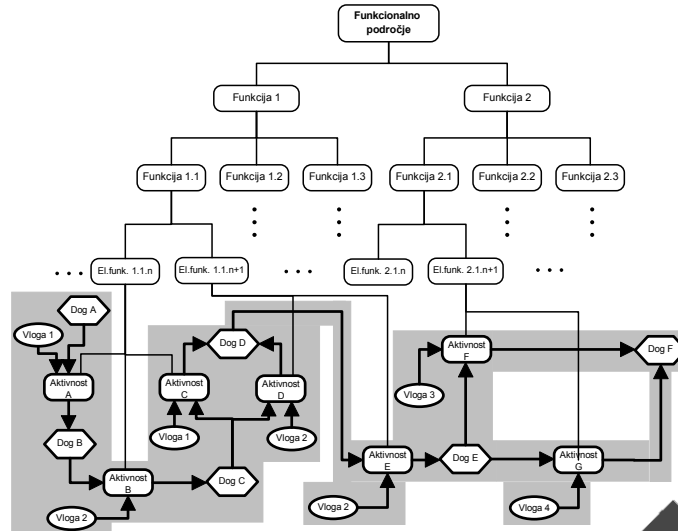
Primer Funkcionalna razgradnja poslovnega sistema TS



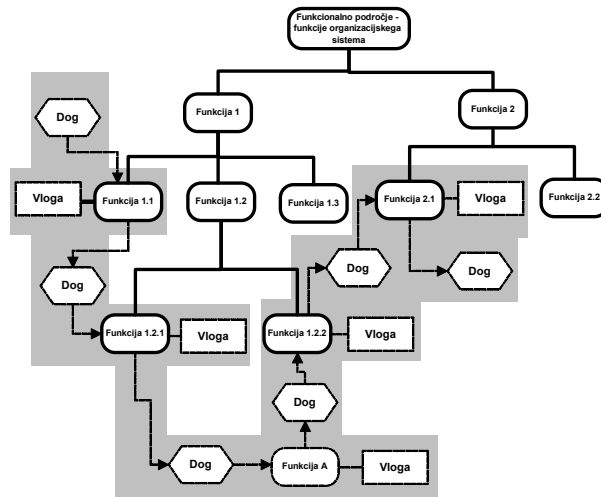
Potek procesa na ravni elementarnih funkcij



Potek procesa na ravni aktivnosti



Potek procesa na ravni funkcij



Informacijski sistem (IS)

Opredelitev IS ◊

- Definicija
 - Informacijski sistem opredelimo kot množico medsebojno odvisnih komponent (strojna oprema, komunikacijska oprema, programska oprema, ljudje), ki zbirajo, procesirajo, hranijo in porazdeljujejo podatke in s tem podpirajo tako temeljne kot tudi odločitvene procese v organizaciji.
- Zanimajo nas formalni in računalniško podprti informacijski sistemi
 - Formalni informacijski sistem ima jasno določene podatke, s katerimi operira, določene postopke za njihovo obdelavo ter jasno definirana pravila
 - Neformalni informacijski sistem je odvisen od implicitnih dogovorov in nedefiniranih pravil.
 - Računalniško podprt informacijski sistem temelji na uporabi računalnikov in informacijske tehnologije.

Značilnosti dobrega IS

- Lastnosti “dobrega” informacijskega sistema:
 - Zagotavlja podatke, iz katerih lahko zaposleni na različnih ravneh v organizaciji pridobijajo informacije, ki jih potrebujejo pri svojem delu.
 - Daje podlago tako za reševanje vsakodnevnih vprašanj kot tudi za izvajanje upravljalških ukrepov ter sprejemanje strateških odločitev.
 - Je usklajen s poslovnim sistemom!

IS - podpora odločanju na vseh ravneh

Strukturiranost vprašanj?

- Primer: podjetje, ki se ukvarja z izvajanjem računalniških tečajev. IS v podjetju daje podlago za reševanje vprašanj, kot so:
 - Vsakodnevna vprašanja:
 - Je Janez Novak prijavljen na tečaj *Windows XP*, ki se prične naslednji teden?
 - Je podjetje *MIX d.o.o.* plačalo račun za svojih sedem udeležencev tečaja iz prejšnjega tedna?
 - Kdo so udeleženci tečaja *Visual Studio*, ki se prične jutri?
 - Upravljalna vprašanja:
 - Je prijavljenih za tečaj *JBuilder* dovolj, da je izvedba tečaja upravičena?
 - Kakšen je bil dobiček s tečajem, ki je bil izveden v Mariboru?
 - Kateri tečaji so bili v zadnjem letu najbolj donosni?
 - Strateška vprašanja:
 - Bi bilo smiselno dvigniti cene tečajev?
 - Je smiselno pripravljati nadaljevalne tečaje?
 - Informatika je v krizi. Je smiselno razmišljati o dodatni dejavnosti?

Vrste IS

- Med osnovne vrste IS štejemo:
 - Transakcijske IS (*TPS-Transaction Processing System*)
 - Upravljalne (poslovodne) IS (*MIS-Management Information System*)
 - Direktorske IS (*ESS-Executive Support System*)
 - Odločitvene IS (*DSS-Decision Support System*)
 - Ekspertne IS (*EIS-Expert Information System*)
 - Sisteme za avtomatizacijo pisarniškega poslovanja (*OAS-Office Automation System*)
 - Sisteme za podporo delovnim procesom (*WfS-Workflow Management System*)

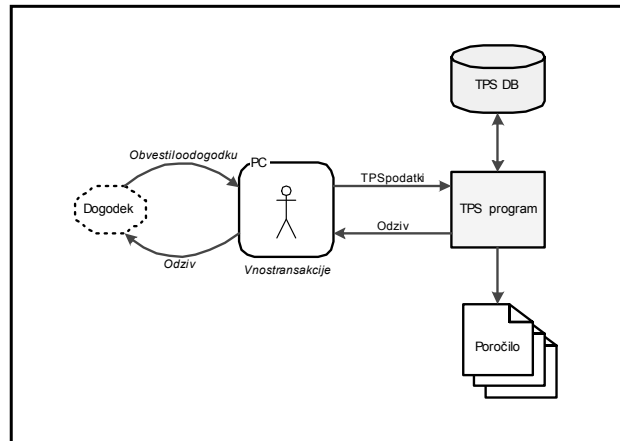
Transakcijski IS (TPS)

- Transakcijski IS (TPS - *Transactional Processing System*) so namenjeni zajemu in hranjenju podatkov o dnevni operacijah, imenovanih tudi transakcije.
- Transakcija je poslovni dogodek, ki generira ali spremeni podatke v podatkovni bazi informacijskega sistema.
- Primeri TPS:
 - Sistem za rezervacijo kart
 - Sistem za zajem naročil
 - Sistem za računanje plač
 - Sistem za vodenje računov
 - Sistem za vodenje evidence gradiva v knjižnici itd.
- TPS so v pomoč pri izvajanju in sledenju vsakodnevnih operacij v poslovnem sistemu, včasih pa nudijo podporo tudi pri odločanju v sklopu izvajanja transakcij (primer: iskanje najprimernejšega leta za izbrano destinacijo).
- So najstarejši sistemi. Prvi so obstajali že v petdesetih letih, predvsem v računskih centrih večjih podjetij.

Razvoj TPS

- TPS gradimo na osnovi podrobnih specifikacij o tem:
 - kako naj se transakcije izvajajo (postopek),
 - kakšne podatke zajemamo o transakcijah, v kakšni obliki naj bodo zajeti ter katerim pravilom, predpisom in ciljem organizacije morajo ustrezati (struktura).
- TPS navadno podpirajo visoko strukturirane procese. Večinoma vključujejo uporabnika, obstajajo pa tudi popolnoma avtomatizirani sistemi (npr. ATM). Nekateri avtomatizirajo celo odločitvene procese v zvezi s transakcijami (npr. iskanje najboljše letalske karte glede na določene predpogoje)
- Dober TPS preveri vsako transakcijo glede na možne predvidene napake (npr. napačen vnos ipd.)
- TPS so za organizacije ključnega pomena. Izpad TPS lahko pomeni hude težave. Varnost TPS zagotavljamo z ustreznimi *backup & recovery* postopki.

Arhitektura TPS



Procesiranje transakcij v TPS

- Glede na način procesiranja transakcij ločimo med:
 - Interaktivnim procesiranjem (*on-line processing*) in
 - Paketnim procesiranjem (*batch processing*).
- V primeru interaktivnega procesiranja je vsaka transakcija procesirana takoj. Uporabnik je v interakciji s sistemom, poda podatke o transakciji, po njenem procesiranju pa dobi obvestilo o izvedbi. Uporabnik je na voljo, ko so za transakcijo potrebni dodatni podatki.
- Pri paketnem procesiranju se podatki o transakcijah najprej zbirajo, kasneje pa procesirajo. Procesiranje se izvede za cel paket transakcij in je lahko sproženo avtomatsko, ob določenem času ipd.
- Današnji TPS večinoma zajemajo obe vrsti procesiranja. Interaktivno procesiranje je veliko bolj občutljivo na odzivnost.

Celoviti informacijski sistemi

- Celoviti IS (*ERP - Enterprise Resource Planing*) predstavljajo višji nivo TPS.
 - Združujejo procesiranje transakcij z različnih funkcionalnih področij.
 - Dajejo enotno podatkovno bazo, iz katere lahko pooblašeni uporabniki pridobijo katerekoli podatke, ki so jim v pomoč pri odločanju znotraj organizacije.
- Uvajanje ERP sistemov je zelo težko. Podjetja ERP sistemov navadno ne gradijo sama, temveč jih kupijo.
 - Dragi sistemi (SAP, Baan,...)
 - Visoki stroški uvajanja (dragi svetovalci)
 - Nizka vpeljana funkcionalnost
 - Zahteva veliko prilagajanja (prenovitev PP)

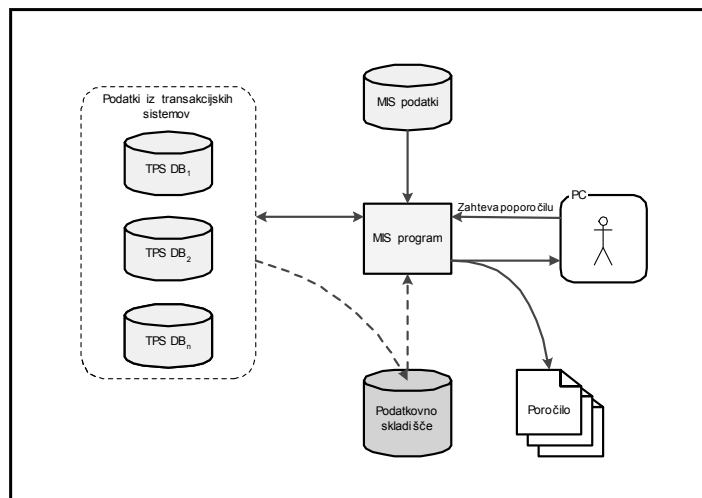
Upravljalni informacijski sistemi

- Upravljalni IS (*MIS - Management Information Systems*) so namenjeni vodstvenim delavcem. Iz TPS jemljejo podatke ter jih oblikujejo (agregirajo) v poročila, ki so v pomoč pri upravljanju organizacije.
- MIS so se pojavili tudi zaradi pomanjkljivosti TPS, ki so sicer izboljšali procesiranje transakcij, za upravljanje pa niso dali veliko informacij.
- MIS se veliko uporabljajo, predvsem v kontrolne namene. Včasih so v pomoč tudi pri planiranju in organizaciji dela.
- MIS se ne ukvarjajo z vsakodnevnimi operacijami, temveč z aktivnostmi, ki so namenjene njihovu

UIS

TIS

Arhitektura upravljaljskih IS



Direktorski informacijski sistemi

- Direktorski IS (*EIS - Executive Information Systems*) so poseben primer upravljaljskih sistemov. V primerjavi s klasičnimi MIS:
 - so bolj interaktivni (MIS omogočajo *monitoring* na osnovi vnaprej izdelanih poročil.)
 - so bolj prilagodljivi različnim poslovnim situacijam (MIS so v pomoč pri iskanju odgovorov na vnaprej znana vprašanja, EIS pa s posebnimi mehanizmi podpirajo reševanje poljubnih vprašanj *managementa* (npr. *iskanje v globino*).
 - uporabljajo vmesnike, ki so enostavni in primerni za delavce na vodilnih položajih (ti navadno nimajo veliko izkušenj z računalniki)
 - dajejo poudarek na enostavnih vmesnikih ter učinkoviti predstavitvi podatkov.

Odločitveni informacijski sistemi

- Odločitveni IS (*DSS - Decision Support Systems*) so interaktivni sistemi, ki na osnovi podatkov, orodij za njihovo obdelavo ter modelov omogočajo odločevalcem, da se lažje odločajo v situacijah, ki niso predvidene in formalizirane.
- Tradicionalni DSS
 - omogočajo interaktivno reševanje odločitvenih situacij,
 - delujejo na osnovi modelov odločanja
 - nudijo posebne mehanizme za analizo alternativnih možnosti (npr. kaj če analiza)
- DSS lahko služijo kot pomoč pri reševanju ponavljajočih poslovnih situacij ter pri reševanju specifičnih, enkratnih situacij.

Odločitveni informacijski sistemi

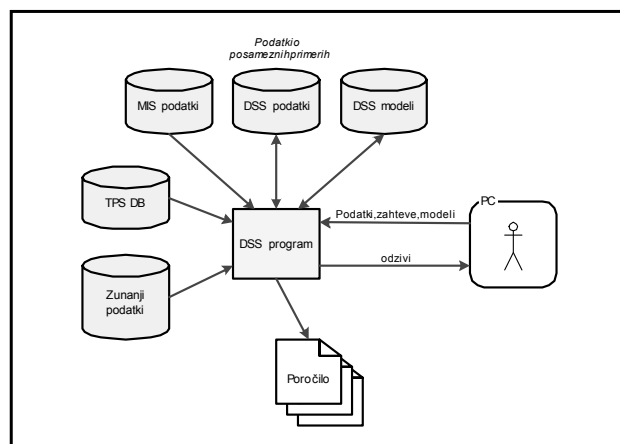
Primer

- Primeri DSS
 - Sistem za pomoč zavarovalniškemu agentu pri sklepanju zavarovanj (del odločitvene situacije, ki je strukturiran, rešuje sistem, nestrukturiran del zahteva odločevalca)
 - Pomoč pri trženju - sistem uporabniku daje na voljo razne interne podatke in podatke, pridobljene iz raznih zunanjih virov. Na voljo ima modele, ki se nanašajo na zadeve, kot so: učinkovitost trženja, strategije konkurence, uporabniško dožemanje izdelkov ipd. Uporabnik lahko s spreminjanjem parametrov v modelih analizira različne alternativne možnosti.

Presek med MIS in DSS

- DSS so nastali kot posledica pomanjkljivosti TPS in MIS na področju reševanja neformaliziranih odločitvenih situacij.
- DSS se delno prekrivajo z EIS, njihova uporaba pa je drugačna. Direktorji, ki uporabljajo EIS, pridobijo podatke o analizah, analitično delo pa opravijo drugi. Uporabniki DSS pa sami opravijo analitično delo. Uporabniki DSS so izkušeni za delo s sistemom, uporabniki EIS za delo s sistemom večkrat zahtevajo pomoč informatikov.
- DSS sistemi zajemajo širok spekter sistemov, od navadnih preglednic (npr. Excel) pa do posebej razvitih sistemov za reševanje določenih poslovnih situacij.
- Novejši pristopi k reševanju odločitvenih situacij zajemajo mehanizme, kot so: OLAP (*On-line Analytical Processing*), *Data Mining* in skupinsko odločanje.

Arhitektura odločitvenih IS



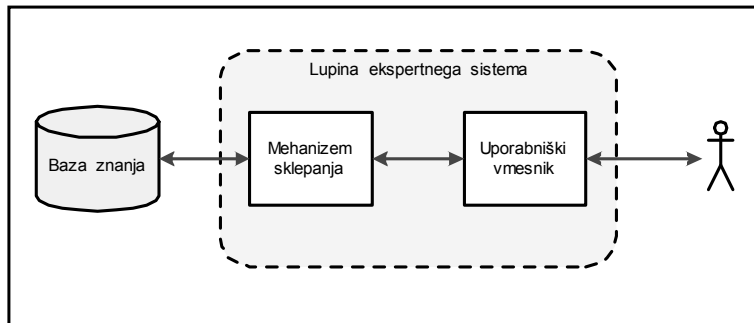
Ekspertni informacijski sistemi

- ES so sistemi, ki se v določenih situacijah obnašajo kot izurjene osebe. Značilnosti:
 - Sposobni so reševanja problemov, ki sicer zahtevajo ekspertno znanje z nekega področja.
 - Znajo obravnavati nepopolne in nezanesljive podatke
 - Delujejo na osnovi baze znanja, ki vsebuje znanje, specifično za problemsko domeno.
 - Z obravnavo vhodnih podatkov glede na znanje, zajeto v bazi znanja, predlagajo rešitev oziroma podajo diagnozo problema.
 - Svoje predloge in diagnoze znajo razložiti.

Ekspertni informacijski sistemi

- ES sestavljajo trije pomembni moduli: Baza znanja (*Knowledge Base*), mehanizem sklepanja (*Inference Engine*) in uporabniški vmesnik (*User Interface*).
- Baza znanja vsebuje znanje, ki je specifično za problemsko domeno. Običajno vsebuje:
 - preprosta dejstva ter pravila, ki določajo oziroma opisujejo relacije v domeni,
 - metode in različne ideje ter heuristiko za reševanje problemov v domeni.
- Mehanizem sklepanja je vmesnik, ki zna uporabljati bazo znanja.
- Uporabniški vmesnik omogoča preprosto komunikacijo med uporabnikom in sistemom. Skupaj tvorita lupino ES, ki je lahko splošna za več ES (bazo znanja ES poljubno zamenjamo, lupina pa ostaja ista).

Arhitektura ekspertnih IS



Predstavitev znanja v ekspertnih IS

- Za zapis znanja v ES se največ uporabljajo t.i. produkcijska pravila, ki jih zapišemo v obliki "if-then".
- Produkcijsko pravilo je pogojni stavek, ki ima lahko različne oblike:
 - če pogoj P potem sklep S
 - če situacija S potem akcija A
 - če pogoj P potem sklep S do določene meje M
 - če pogoj p1 in pogoj p2 potem ni res pogoj p3

Predstavitev znanja v ekspertnih IS (2)

- Predstavitev znanja s pomočjo produkcij ima več dobrih lastnosti:
 - Modularnost: pravila vedno veljajo neodvisno od drugih pravil.
 - Razširljivost: bazo znanja lahko preprosto razširimo.
 - Prilagodljivost: ker so pravila neodvisna med seboj, jih lahko neodvisno spreminjamo.
 - Transparentnost: sistem zna pojasniti, zakaj in kako je prišel do nekega sklepa.

Sistemi za avtomatizacijo pisarniškega poslovanja

- Sistemi za avtomatizacijo pisarniškega poslovanja (*OAS - Office Automation Systems*) vsebujejo orodja za podporo osnovnemu pisarniškim aktivnostim:
 - Izdelava raznih izračunov (TPS),
 - Urejanje dokumentov,
 - Organizacija sestankov,
 - Nadzor nad pisarniškim poslovanjem ipd.
- OAS zajemajo širok spekter orodij:
 - Preglednice (DSS),
 - Urejevalniki besedil,
 - Orodja za pripravo predstavitev
 - Komunikacijski sistemi
 - Telekonferenčni sistemi (*Teleconferencing Systems*),
 - Sporočilni sistemi (*Messaging Systems*),
 - Sistemi za podporo skupinskemu delu (*Groupware Systems*),
 - Sistemi za upravljanje z znanjem (*Knowledge Management Systems*)

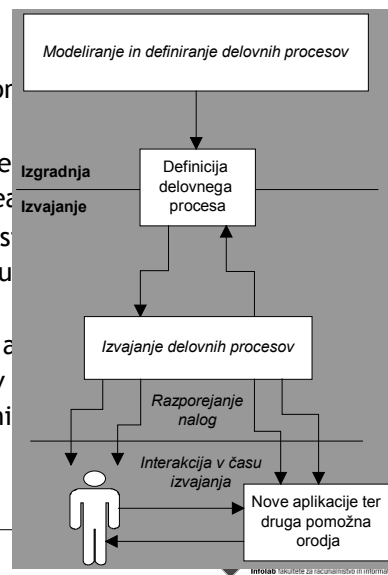
Sistemi za podporo delovnim procesom

- Sistem za podporo delovnim procesom (*WfS - Workflow System*) je sistem, kjer so definirani, krmiljeni, izvajani in nadzorovani delovni procesi ali deli delovnih procesov z uporabo informacijske tehnologije, pri čemer je zaporedje izvajanja aktivnosti v celoti definirano z logičnim zapisom delovnih procesov, ki ga razume ta sistem.
- Upravljanje delovnih procesov zajema:
 - Definiranje
 - Krmiljenje
 - Izvajanje
 - Nadzorovanje

Informacijski sistem

Sistemi za podporo delovnih procesom (WfS)

- WfS podpira štiri funkcije:
 - Funkcija izgradnje sistema, ki por modeliranje delovnih procesov.
 - Funkcija izvajanja delovnih procesov v re primerkov delovnih procesov v re
 - Funkcija razporejanja nalog in sis pomeni razporejanje nalog med u udeležence procesov.
 - Funkcija interakcije v času izvaja sodelovanje končnih uporabnikov tehnologije pri izvedbi posamezni procesa.



Sodobni informacijski sistemi

- Področje IT se hitro razvija. Sodobni IS ne ustrezajo več klasičnim kategorijam, temveč pogosto podpirajo funkcionalnosti, ki pripadajo več kategorijam.
- Kljub temu, da kategorizacija IS ne ustreza več dejanskemu stanju ali pa obstajajo različni pogledi nanjo, se izkaže koristna, saj poudarja karakteristike posameznih kategorij, med katerimi so mnoge take, ki jih kaže upoštevati v vsakem IS.

Mesto IS v poslovnem okolju

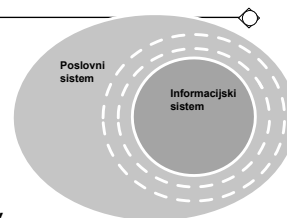


Mesto IS v poslovnem okolju - Informacijska tehnologija - IT



- Informacijska tehnologija označuje:
 - programsko opremo (*software*) in
 - strojno opremo (*hardware*),
 ki se uporablja za podporo delovanju informacijskega sistema.
- Strojna oprema se nanaša na naprave in drugo fizično opremo:
 - delovne postaje, strežniki,
 - tiskalniki,
 - omrežje,
 - UPS ipd.
- Programska oprema so računalniški programi, ki sprejemajo vhodne podatke in vodijo delo strojne opreme.
 - sistemska programsko opremo (npr. operacijski sistem)
 - uporabniška oprema (npr. urejevalnik besedil, preglednice, specializirana oprema, namenjena podpori določeni poslovni funkciji, itd.)
- Med informacijsko tehnologijo štejemo tudi t.i. tehnologijo papir in pisalo, ki je v uporabi v računalniško nepodprtih informacijskih sistemih.

Večanje preseka med PS in IS



- Presek med poslovnim sistemom in podpornim informacijskim sistemom se večja.
- Delo se izvaja v poslovnem sistemu, podatki o tem pa se zbirajo in obdelujejo v informacijskem sistemu.
- Primerjava med različnima poslovnima sistemoma:
 - poslovni sistem, katerega osnovni namen je gojenje trt in prodaja grozdja ter sistem, ki podpira *Izbirni postopek* za vpis na visokošolske zavode v Sloveniji
- Večanje preseka je posledica hitrega razvoja informacijskih tehnologij

Obraznava poslovnih sistemov

- WCA - *Work Centered Analysis framework*
 - WCA predstavlja splošno shemo, ki daje začetni okvir za proučevanje poslovnih ter informacijskih sistemov.
 - WCA je dobila ime po tem, ker poudarja potrebo poslovnih uporabnikov po dobrem razumevanju poslovnega sistema, da bi se lahko odločali o potrebi po gradnji, izboljšavah, ali prenovitvi informacijskih sistemov.
 - WCA združuje ideje različnih disciplin, na primer:
 - Upravljanje kakovosti
 - Prenovitev poslovnih procesov
 - Teorija sistemov ipd.

Gradniki sheme WCA

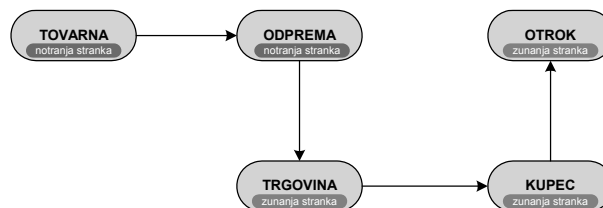
- Osnovni gradniki sheme WCA zajemajo:
 - Notranje in zunanje stranke (uporabniki poslovnega sistema)
 - Izdelke (proizvode, produkte) poslovnega sistema
 - Aktivnosti (korake) poslovnega sistema
 - Udeležence poslovnega sistema
 - Podatke (informacije), ki jih poslovni sistem kreira ali uporablja
 - Tehnologijo, ki jo poslovni sistem uporablja

Gradniki sheme WCA (2)



Stranke (gradnik WCA)

- Notranje stranke
- Zunanje stranke
- Kdo so stranke tovarne, ki izdeluje otroške igrače?



Izdelek (gradnik WCA)

- Izdelek je rezultat oziroma izhod poslovnega sistema.
- Izdelek je lahko:
 - fizičen objekt
 - storitev
 - podatek
- Karakteristike izdelka:

- Cena	- Odzivnost
- Kakovost	- Zanesljivost
- Dostopnost	- Ustreznost standardom



Poslovni proces (gradnik WCA)

- Ponovitev
 - Poslovni proces je povezana skupina korakov oziroma aktivnosti, ki se izvajajo v poslovnem sistemu in posredno ali neposredno vplivajo na dodano vrednost pri uresničevanju skupnega cilja poslovnega sistema.
 - Aktivnost je je majhna naloga, korak ali operacija znotraj procesa in je navadno najmanjša enota, ki jo določimo pri obravnavi poslovnega procesa.
 - Aktivnosti so časovno in prostorsko povezane, imajo začetek in konec ter vhodne in izhodne elemente.
- Poslovni proces je ključen, vendar ne edini element, ki ga obravnavamo po WCA shemi.



Upravljanje z znanjem

- Znanje je strateška pridobitev vsake organizacije oziroma njeno premoženje.
- Uspešnost organizacije postaja vse bolj odvisna od njene sposobnosti upravljanja z znanjem.
- Področje, ki se ukvarja z upravljanjem znanja, lahko delimo po pomenu obravnave:
 - Upravljanjem z znanjem kot objektom, ki ga lahko identificiramo in z njim upravljamo v sklopu računalniško podprtih sistemov (računalniška in informacijska znanost).
 - Upravljanjem z znanjem kot sklopom izkušenj, sposobnosti in *know-how*-a posameznikov ali organizacije, ki je dinamično in se konstantno spreminja (filozofija, sociologija in psihologija).

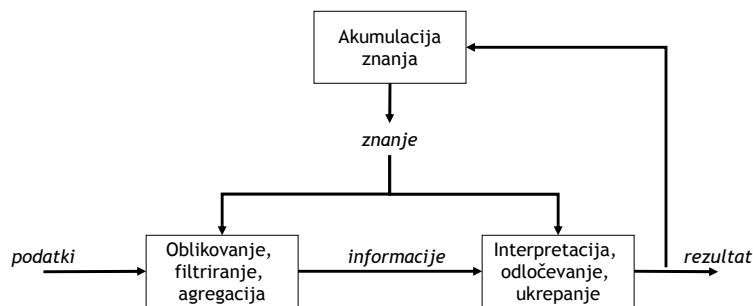


Podatki (gradnik WCA)

- Podatki, s katerimi imamo opravka v poslovnem sistemu, lahko zavzamejo številne oblike: tekst, številke, slike, zvok, video zapis itd.
- Podatki lahko prihajajo od zunaj ali v sistemu nastajajo.
- Kakšno je razmerje med podatki, informacijo in znanjem?



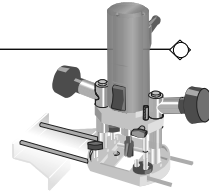
Podatek, informacija, znanje



Vrste znanja

- Znanje lahko delimo na:
 - tacitno ali skrito znanje in
 - eksplicitno znanje.
- Eksplicitno znanje je formalizirano znanje, ki ga je moč razmeroma enostavno izraziti, običajno v obliki principov, postopkov, dejstev, likov, pravil, formul itd. Sčasoma postane rutinsko in prevzame značaj podatkov
- Tacitnega znanja ni enostavno izraziti niti videti. Je precej subjektivno in prepleteno z vedenjem in časom. Obsega izkušnje, ideale, čustva, intuicijo in notranji vpogled. Deli se na *tehnično znanje (know how)* in *zaznavno ali kognitivno znanje*.

Tehnologija (gradnik WCA)



- Tehnologija so orodja, ki bodisi neposredno izvajajo določene aktivnosti znotraj poslovnega sistema ali pa so v pomoč udeležencem pri izvajanju njihovega dela.
- Posebna vrsta tehnologije, ki je pri obravnavi informacijskih sistemov najpomembnejša, je informacijska tehnologija.
- Informacijska tehnologija označuje programsko (*software*) in strojno (*hardware*) opremo, ki se uporablja za podporo delovanju informacijskega sistema.

3 SPLOŠNO O RAZVOJU INFORMACIJSKIH SISTEMOV

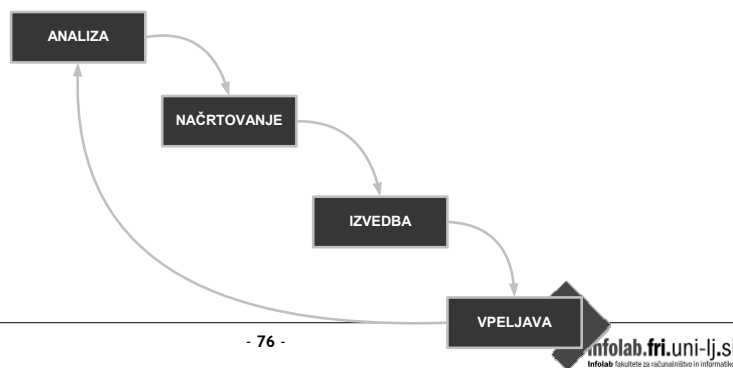
- Življenjski cikli razvoja: Zaporedni model, Iterativni model, prototipiranje, Inkrementalni model, Kombinirani model
- Metodologija ali proces, jezik, metoda ali tehnika
- Zgodovina razvoja informacijskih sistemov
- CASE orodja, orodja za podporo pri razvoju informacijskih sistemov

Življenjski cikli razvoja IS

- Kot večina razvojnih procesov sledi tudi razvoj IS določenemu življenjskemu ciklu, oziroma razvojnemu modelu, ki določa zaporedje faz razvoja.
- Razvojni modeli IS zajemajo analizo, načrtovanje, izvedbo ter vpeljavo. Med seboj se razlikujejo predvsem po podrobnejši delitvi faz na aktivnosti ter v zaporedju in načinu njihovega izvajanja.

Zaporedni ali slapovni model

- Zaporedni ali slapovni model (waterfall model)
 - Najstarejši razvojni model, značilen za prve oblike strukturnega pristopa
 - Faze si sledijo zaporedno
 - Vračanje nazaj ni mogoče



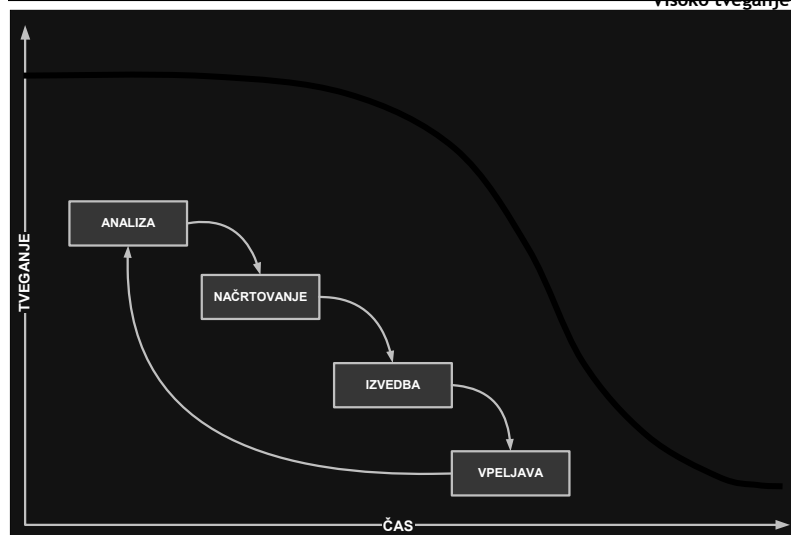
Zaporedni ali slapovni model (2)

Slabosti

- SLABOSTI zaporednega modela
 - Zahteve nikoli niso statične in se spreminjajo.
 - Posamezne faze ne moremo preprosto zaključiti, potrebno je vračanje nazaj.
 - Zaporedni model ne dopušča vračanja nazaj → razvit sistem lahko ne ustreza dejanskim zahtevam
 - Tveganje, da sistem ne ustreza zahtevam je visoko vse do zadnje faze razvoja

Zaporedni ali slapovni model (3)

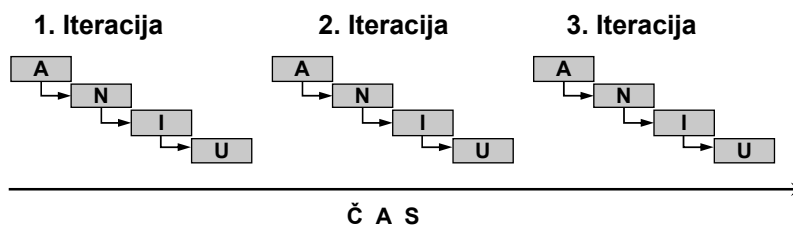
Visoko tveganje



Iterativni model

- Razvit kot odziv na pomanjkljivosti slapovnega pristopa.
- Pri iterativnem pristopu izvajamo korake slapovnega pristopa v več iteracijah.
- V vsaki iteraciji razvijemo določen del funkcionalnosti celotnega sistema.
- V začetnih iteracijah razvijemo najbolj tvegane dele sistema.
- Sistem se razvija inkrementalno.

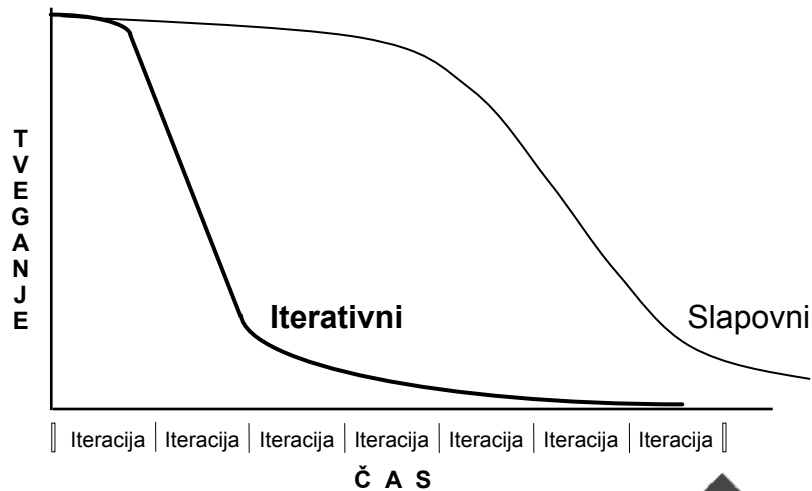
Iterativni model (2)



- Najbolj tvegane so začetne iteracije - najprej razvijemo najbolj tvegan del sistema
- Rezultat vsake iteracije je izvršljiv dodaten del celotnega sistema
- Vsaka iteracija vključuje povezovanje v celoten sistem in preizkušanje

Iterativni razvoj (3)

Nizko tveganje



Iterativni razvoj (4)

Prednosti

- Prednosti iterativnega razvoja (proti zaporednemu):
 - Najbolj tvegani deli so razrešeni še preden postane investicija velika
 - Začetne iteracije omogočijo zgodnje povratne informacije s strani uporabnikov
 - Preizkušanje in povezovanje v sistem sta nepretrgana
 - Ciljni mejniki omogočajo kratkoročno osredotočenje
 - Napredek merimo z ocenjevanjem izvedenega dela
 - Možna je predaja izvedenega dela projekta še preden je dokončan celoten projekt

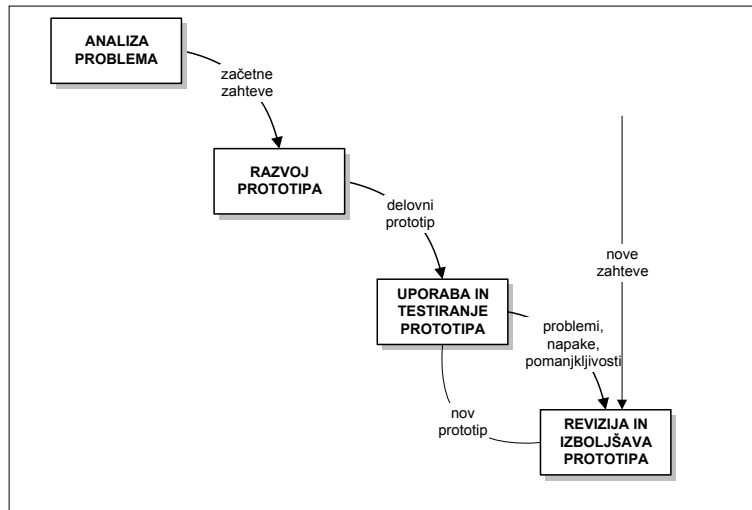
Prototipni razvoj

- Pojavi se z iterativnim modelom
- Danes se uporabljajo pri večini razvojnih modelov
- Obstaja tudi poseben prototipni model
- Prototipni model temelji na izdelavi prototipov
- Prototip označuje predhodno izdelane in navadno nepopolne verzije sistema.
- Uporaba v različnih fazah razvoja.
- Za izdelavo prototipov so bila razvita posebna razvojna okolja.

Prototipni razvoj (2)

- Prototipi se lahko uporabljajo:
 - kot del specifikacije sistema, za pridobitev jasnejše podobe bodočega sistema in se v nadaljevanju zavržejo,
 - kot osnova za izdelavo produkcijskega sistema (npr. *Rapid Application Development* - RAD).
- Vrste prototipov:
 - Funkcionalni
 - Tehnološki

RAD - Rapid Application Development



Inkrementalni model

- Vsebuje prvine iterativnega modela
- Sistem razbijemo na neodvisne dele - razvoj posameznega dela pomeni poseben projekt
- Iteracija iz iterativnega modela označuje sklop opravil znotraj projekta, inkrement iz inkrementalnega modela pa zaključuje sklop sistema

Inkrementalni model (2)



Kombinirani razvojni model



Kombinirani model (2)

- Zasnovan na osnovi zaporednega modela
- Omogoča vračanje v predhodne faze
- Nudi hrbtnico - potrebno pri večjih projektih
- V praksi se veliko uporablja - je zelo brizu naravnemu procesu razvoja - nudi osnovno zaporedje ter dopušča poljubna prehajanja med fazami

Metodologije razvoja IS

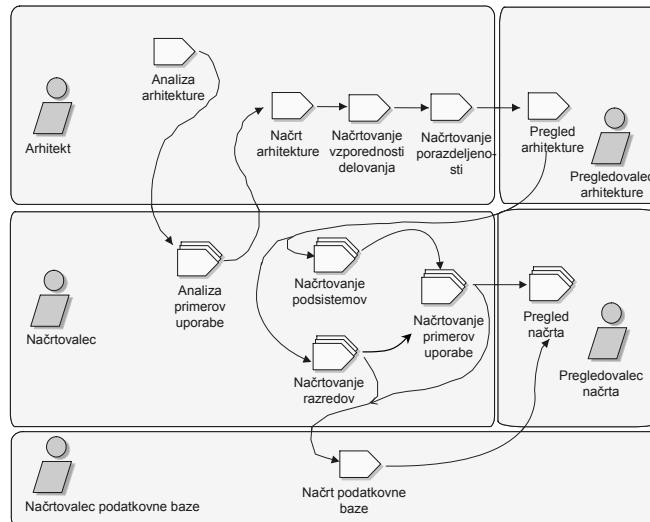
- Pristopi k razvoju IS so se oblikovali skozi leta in izkušnje. Nastale so metodologije razvoja IS.
- Metodologija razvoja IS navadno sledijo izbranemu pristopu in natanko predpisujejo korake, postopke, tehnike, izdelke in orodja za njihovo izdelavo v posameznih korakih razvoja IS.
- Metodologije so prežete s filozofijo njihovih snovalcev.
- Mnoge metodologije so nastale v raziskovalnih krogih, mnoge pa so rezultat praktičnih izkušenj posameznih podjetij s področja razvoja IS.

Metodologije razvoja IS (2)

Osnovni gradniki metodologije



Primer postopka iz metodologije RUP



Primeri metodologij razvoja IS

- *IE - Information Engineering* (Strukturni pristop/ James Martin/ 1981)
- *CASE *Method* (Strukturni pristop/ Richard Barker, Oracle)
- *SSADM - Structured System Analysis and Design Method* (Strukturni pristop/ CCTA-Central Computing and Telecommunications Agency (1981) razvije metodologije za vladne organizacije/ CCTA predlaga SSADM kot standard)
- *OMT - Object Method Technique* (Objektni pristop/ Jim Rumbaugh)
- *RUP - Rational Unified Process* (Objektni pristop/ Rational)

Značilnosti sodobnih metodologij

- Sodobne metodologije razvoja IS so zaznamovane z:
 - Z naglim naraščanjem procesne moči vseh vrst računalnikov,
 - Z integracijo poslovnih procesov ter poslovnih IS,
 - S porazdeljenim procesiranjem in razvojem računalniških mrež,
 - Z bogato ponudbo standardnih aplikativnih rešitev,
 - Z razvojem in naglo uveljavljanje računalniških orodij za razvoj in projektiranje IS.

Značilnosti sodobnih metodologij (2)

- Sodobna metodologija razvoja IS upošteva naslednje zahteve:
 - Zajemati mora celoten življenjski cikel IS in pripadajoče programske opreme, in ne zgolj posameznih faz.
 - Omogočati mora sistematičen prehod v naslednjo fazo.
 - Omogočati mora preverjanje pravilnosti procesa skozi vse faze življenjskega cikla IS.
 - Podpirati mora skupinsko delo na projektu razvoja IS ter omogočati uporabo sodobnih metod organizacije in vodenja projektov.

Značilnosti sodobnih metodologij (3)


- Sodobna metodologija razvoja IS upošteva naslednje zahteve (nadaljevanje):
 - Biti mora uporabna za čim širši spekter računalniških projektov.
 - Biti mora dovolj enostavna za priučitev.
 - Omogoča mora uporabo čim širšega spektra avtomatiziranih orodij za povečanje produktivnosti posameznikov in celotne skupine.
 - Omogočati mora dokumentiranje in spremljanje razvoja IS skozi vso njegovo življenjsko dobo.

Razvojni sistemi

CASE orodja 

- Danes so na voljo različna orodja za podporo posameznim aktivnostim razvoja IS.
- CASE orodja: *Computer Aided Software Engineering*
 - *Upper CASE*: orodja, ki podpirajo aktivnosti prvih faz razvoja IS: poslovno modeliranje, vzpostavitev okvirjev projekta, zajem informacij, konceptualno modeliranje, analiza in načrtovanje IS.
 - *Lower CASE*: orodja, ki so specializirana za podporo izvedbeni IS ter njegovemu vzdrževanju: generiranje programske kode, podatkovne baze, baznih sprožilcev in baznih procedur.
 - *I-CASE*: Skupina integriranih orodij, ki podpirajo vse faze življenjskega cikla razvoja IS.

Razvojni sistemi (2)

SUPB 

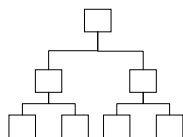
- Sistemi za upravljanje s podatkovnimi bazami (SUPB), npr: *Oracle, MS SQL Server, Ingress, DB2, Inormix, Sybase* itd.
- Sistemi SUPB omogočajo delo s podatki (hranjenje, iskanje, obdelava).
- Vrste podatkovnih baz:
 - Hierarhične PB
 - Mrežne PB
 - Relacijske PB
 - Objektne PB

4 EMRIS - Enotna metodologija razvoja IS

- Osnovne značilnosti
- Faze
- Tehnike
- Prednosti in slabosti

Strukturni pristop

- Eden prvih sistematičnih pristopov k razvoju IS
- Zgleduje se po standardnih postopkih razvoja tehničnih izdelkov: aktivnosti si sledijo zaporedno.
- Izoblikoval se je konec 60 in v začetku 70 let.
 - Razlog: uvedba discipliniranega izvajanja analize in načrtovanja.
 - Cilj: zmanjšanje stroškov izgradnje in uvajanja IS.
- Pristop *Top Down*



Strukturni pristop (2)

Strukturni pristop ◊

- Najpomembnejše tehnike:
 - Diagrami podatkovnih tokov
 - Funkcionalna dekompozicija
 - Diagrami entiteta-razmerje
 - ...

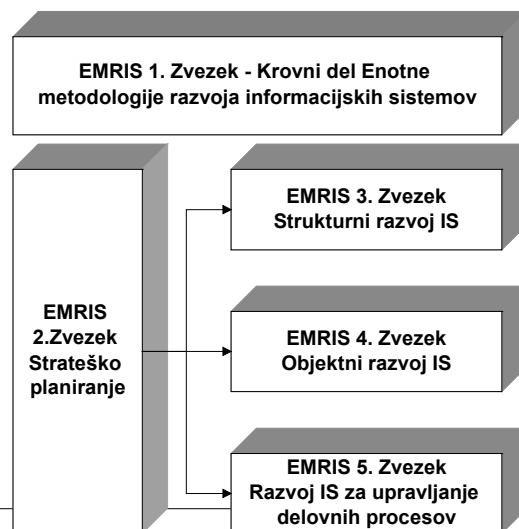
Kaj je EMRIS

- Enotna metodologija razvoja informacijskih sistemov
- Metodologija, ki obsega:
 - opis procesa razvoja
 - opis metod in tehnik uporabljenih v različnih razvojnih fazah
 - Prikaz primerov izdelkov

Komu je EMRIS namenjen?

- vodjem projektov strateškega planiranja ter razvoja IS
- vodjem in članom skupin za zagotavljanje kakovosti
- izvajalcem strateškega planiranja in razvoja IS

Zgradba EMRIS



Strateško planiranje

- Razvoj IS organizacije se po EMRIS začne s fazo strateškega planiranja informatike.
- Cilji strateškega planiranja so:
 - Povezati razvoj IS s poslovno strategijo organizacije.
 - Izboljšati komunikacijo med vodstveno strukturo in informatiki.
 - Načrtovati pretok informacij in procesov (zmanjša obseg nepotrebne dela, nekonsistentnost in redundanco podatkov, poveča kakovost in točnost informacij).

Strateško planiranje (2)

- Cilji strateškega planiranja (nadaljevanje):
 - Zmanjšati stroške in skrajšati čas, potreben za razvoj aplikacij.
 - Predlagati optimalno zaporedje nadaljnjih korakov pri planiranju in razvoju IS.
 - Pripraviti vsa potrebna izhodišča za pomoč pri nadaljnjih korakih informatizacije vse do izdelave aplikativnih sistemov.
 - Zagotoviti uporabo standardov za enotne tehnološke rešitve.
 - Pokazati na organizacijske probleme pri uvajanju informacijske podpore in predlagati organizacijske rešitve za doseg racionalnejše uporabe informacijske podpore.

Strateško planiranje (4)

- Pri strateškem planiranju obravnavamo naslednje elemente:
 - cilje, usmeritve, probleme in kritične dejavnike uspeha (KDU), organizacijske enote, geografske lokacije, funkcionalna področja, funkcije in postopke, delovne procese, entitete, informacijsko tehnologijo, kadre itd.
- Izdelki, ki pri tem nastanejo, so:
 - Seznam strateških elementov
 - Pregledni model,
 - Analiza obstoječega stanja IS,
 - Načrt IT in
 - Plan razvoja IS.

Strateški elementi

Organizacijske enote

Funkcije

IT

Delovni procesi

Entitete

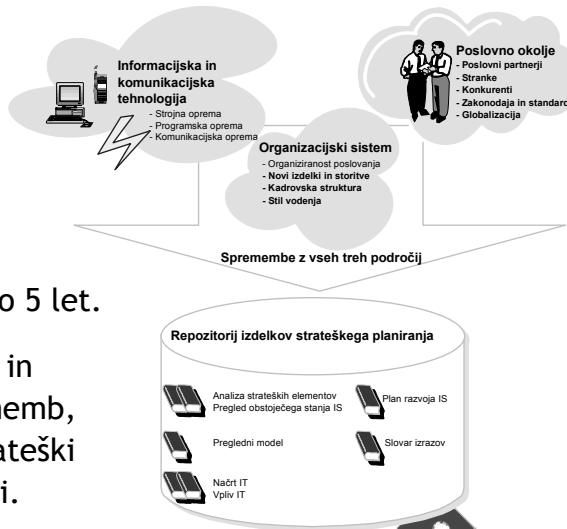
Kadri

Strateško planiranje (5)

- Izdelava strateškega plana traja približno od 3 do 6 mesecev.
- Pri izdelavi sodelujejo:
 - Zunanji svetovalci
 - Metodologi (informatike izven organizacije)
 - Ključni uporabniki
 - Člani vodstvene skupine organizacije
- Strateški plan je potrebno osveževati! Govorimo o skrbništvu strateškega plana

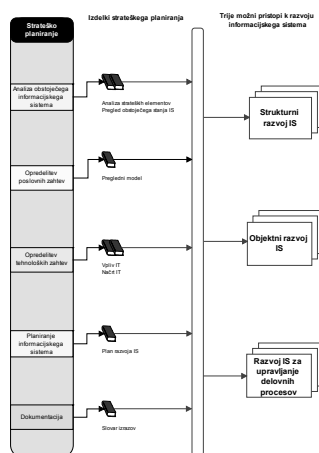
Strateško planiranje (6)

- Strateško planiranje je stalni proces.
- Uresničevanje strateškega načrta traja približno od 2 do 5 let.
- Zaradi zunanjih in notranjih sprememb, je potrebno strateški načrt osveževati.



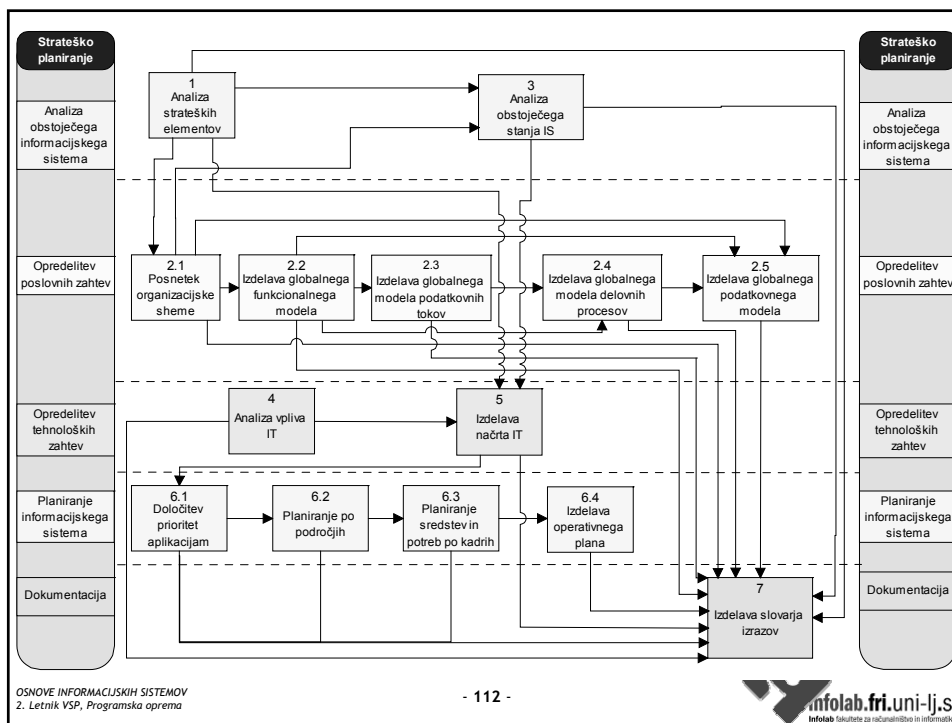
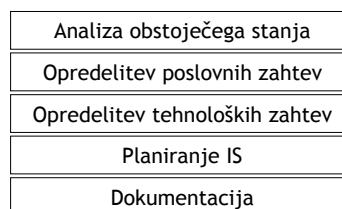
Strateško planiranje (7)

- Izdelki, ki nastanejo v fazi izdelave strateškega načrta, služijo kot vhod v razvoj posameznih informacijskih projektov.
- Strateški načrt je podlaga za organizacijo projektov ter analizo področja, za katerega razvijamo informacijsko podporo.



Strateško planiranje (8)

- Postopek izdelave strateškega planiranja opišemo z naslednjimi aktivnostmi:
 - Analiza obstoječega stanja
 - Opredelitev poslovnih zahtev
 - Opredelitev tehnoloških zahtev
 - Planiranje informacijskega sistema
 - Dokumentacija



Analiza obstoječega stanja

- Namen analize obstoječega stanja je analizirati strateške elemente organizacije, da bo razvit IS v celoti usklajen z njenimi smernicami in cilji ter podati pregled obstoječega stanja IS.
- Opravila:
 - Analiza strateških elementov
 - Analiza obstoječega stanja IS

Analiza obstoječega stanja
Opredelitev poslovnih zahtev
Opredelitev tehnoloških zahtev
Planiranje IS
Dokumentacija

Analiza obstoječega stanja (2)

- Analiza strateških elementov obravnava:
 - Poslanstvo: najvišji strateški element, ki opredeljuje smisel obstoja organizacije.
 - Vizijo: jedrnat zapis zelenih dosežkov, ki jih organizacija s svojim delovanjem skuša doseči.
 - Cilje: taktični (kratkoročni) in strateški (dolgoročni) cilji - zeleno stanje organizacije. Cilj mora biti definiran po obsegu in času, biti mora merljiv
 - Usmeritve: posplošitev ciljev.
 - Probleme: problemi, ki ovirajo delovanje sistema.
 - Kritične dejavnike uspeha (KDU): dejavniki, ki so ključni za zagotovitev uspešnega delovanja organizacije.
- Cilji, problemi in KDU se obravnavajo posebej za PS in IS.

Analiza obstoječega stanja ⁽³⁾

- Podatke o strateški elementih pridobimo s pomočjo:
 - Dokumentov in podatkovnih zbirk,
 - Delovnih sestankov,
 - Vprašalnikov,
 - Intervjujev,
 - Zaključnega sestanka.
- Pomembno je sodelovanje najvišjega vodstva. To je hkrati tudi problem.

Analiza obstoječega stanja ⁽⁴⁾

- V okviru analize obstoječega stanja IS opravimo analizo oz. pregled trenutnega stanja za naslednja področja:
 - strojna oprema,
 - programska oprema,
 - komunikacijska oprema,
 - pregled organiziranosti informatike in kadrov,
 - Informacije na internetu,
 - Analiza vlaganj in stroškov.

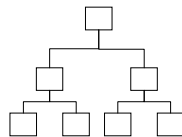
Opredelitev poslovnih zahtev

- Namen opredelitve poslovnih zahtev je doseči čim večjo stopnjo razumevanja dogajanja v organizaciji ali delovnem področju.
- Opravila zajemajo izdelavo naslednjih izdelkov:
 - Organizacijske sheme
 - Globalnega funkcionalnega modela
 - Globalnega modela podat. tokov
 - Globalnega modela del. procesov
 - Globalnega podatkovnega modela
 - Povezovalnih matrik

Analiza obstoječega stanja
Opredelitev poslovnih zahtev
Opredelitev tehnoloških zahtev
Planiranje IS
Dokumentacija

Opredelitev poslovnih zahtev (3)

- Definicija:
 - Organizacijska shema je sestavljena iz grafičnega prikaza organizacijske strukture ter opisa organizacijskih enot.
- Tehnika:
 - Za izdelavo organizacijske sheme uporabljamo funkcionalno dekompozicijo, do največ četrte ravni. Poznamo navpično in vodoravno razgradnjo. Najenostavnejši je prikaz v obliki drevesa.
 - Povezovalna matrika med organizacijskimi enotami in lokacijami



Opredelitev poslovnih zahtev (4)

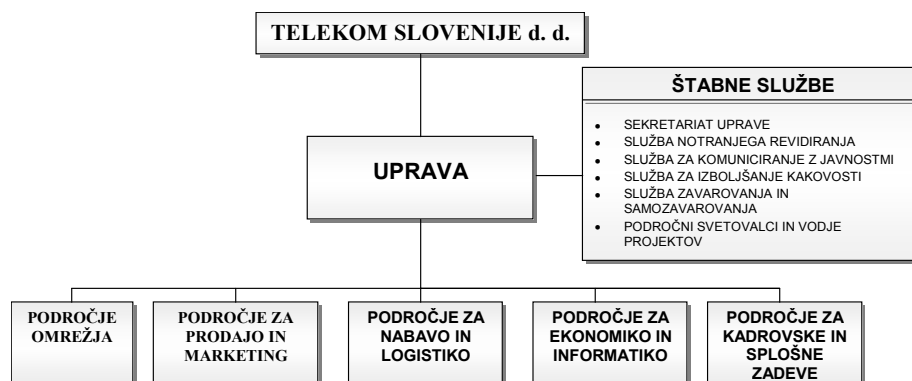
Organizacijska shema 

- Metoda dela:
 - Podatke za izdelavo organizacijske sheme pridobimo s pomočjo:
 - Obstoječe dokumentacije
 - Delovnih sestankov.
 - Pri opisu organizacijskih enot podamo:
 - Kratak opis področja dela organizacijske enote,
 - Podatke o številu zaposlenih ter o izobrazbi članov organizacijske enote

Opredelitev poslovnih zahtev (5)

Organizacijska shema 

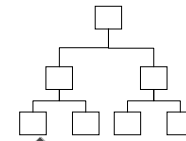
- PRIMER



Opredelitev poslovnih zahtev (6)

Glob. funkcionalni model 

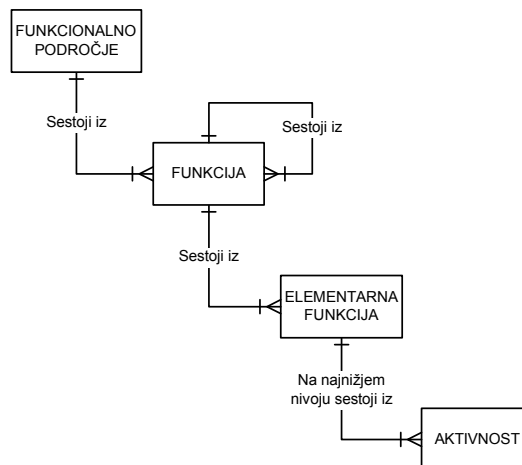
- Definicija:
 - Globalni funkcionalni model je sestavljen iz grafičnega prikaza razgradnje funkcij ter opisa funkcij.
- Tehnika:
 - Za izdelavo globalnega funkcionalnega modela uporabljamo funkcionalno dekompozicijo, do tretje ali četrte ravni. Poznamo navpično in vodoravno razgradnjo. Najenostavnejši je prikaz v obliki drevesa.
 - Povezovalna matrika med funkcijami in organizacijskimi enotami



Opredelitev poslovnih zahtev (7)

Glob. funkcionalni model 

- Funkcionalna področja
- Poslovne funkcije
- Elementarne funkcije
- Aktivnosti



Opredelitev poslovnih zahtev (8)

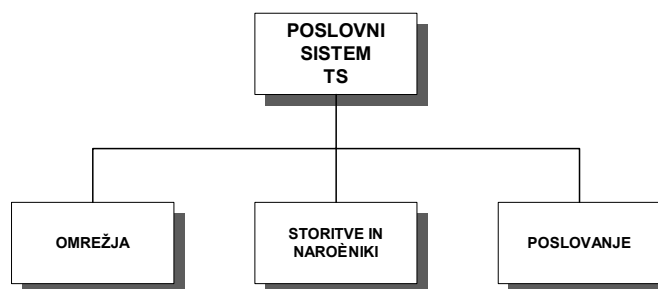
Glob. funkcionalni model 

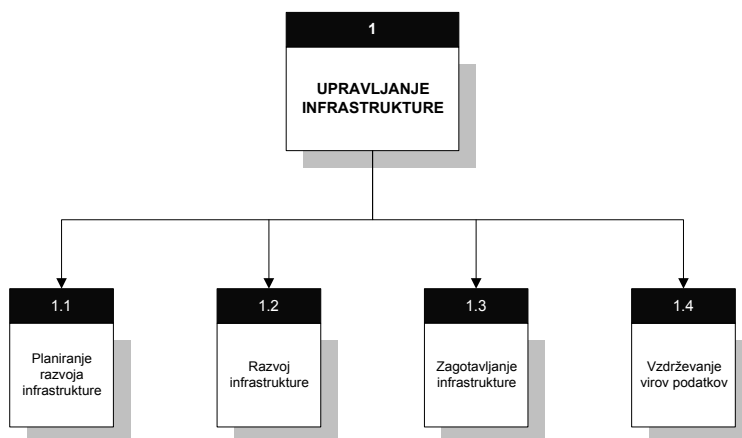
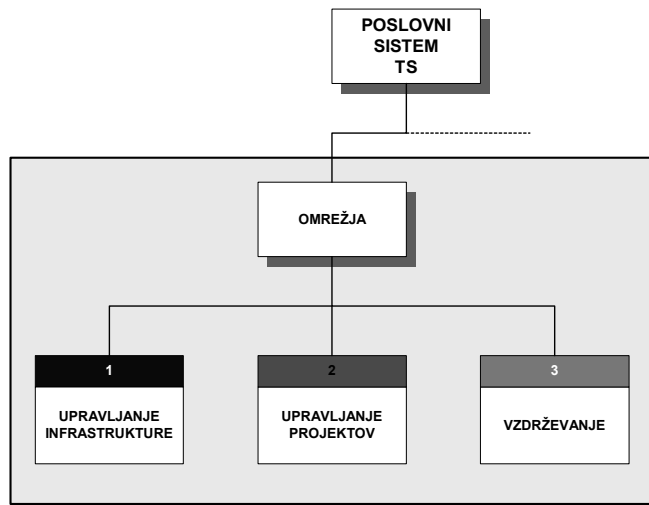
- Metoda dela:
 - Pri izdelavi globalnega funkcionalnega modela se opiramo na naslednje vire:
 - Obstoječa dokumentacija
 - Organizacijska shema
 - Gradivo delovnih sestankov
 - Problemi:
 - Terminologija
 - Neuravnoteženost med elementi posameznih vej drevesa razgradnje

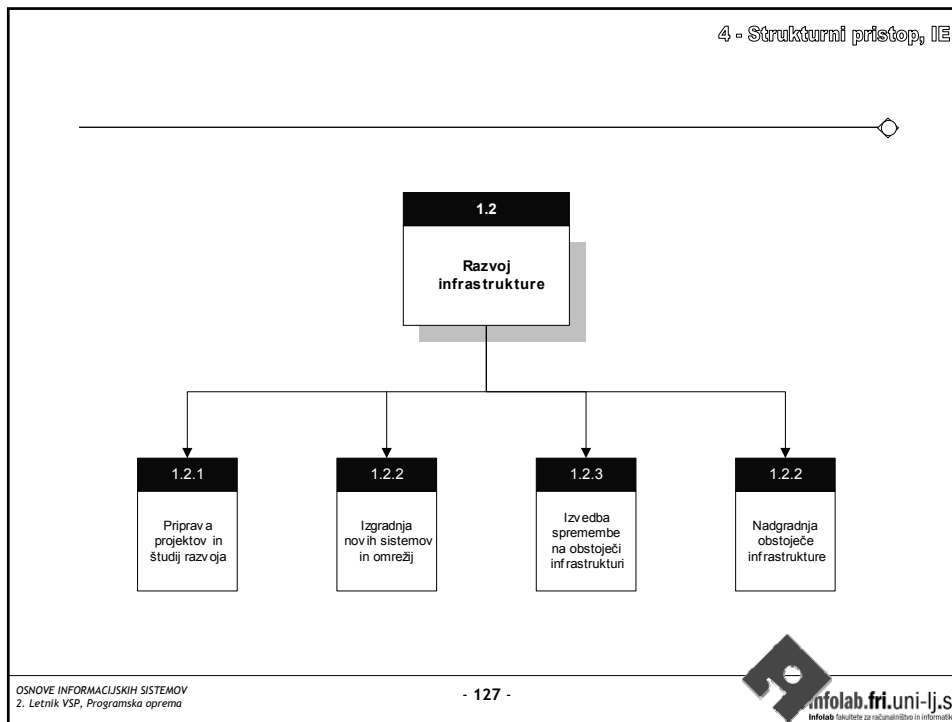
Opredelitev poslovnih zahtev (9)

Glob. funkcionalni model 

- PRIMER



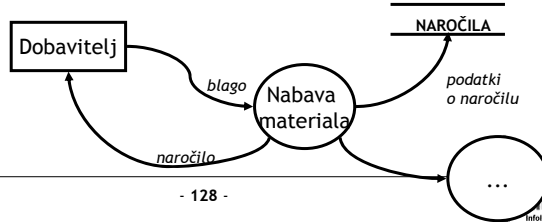




Opredelitev poslovnih zahtev (10)

Glob. model podatkovnih tokov

- Definicija:
 - Globalni model podatkovnih tokov zajema diagram podatkovnih tokov ter opis elementov diagrama. Z njim pokažemo medsebojno sodelovanje funkcij znotraj organizacije ter sodelovanje z okoljem.
- Tehnika:
 - Za izdelavo globalnega modela podatkovnih tokov uporabljamo tehniko DFD - diagram podatkovnih tokov.



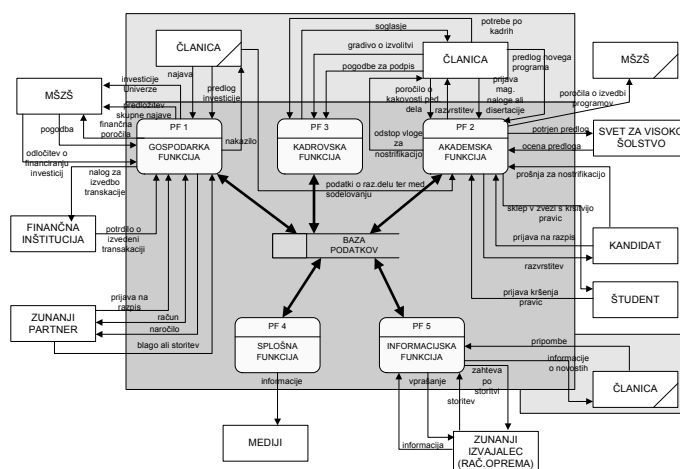
Opredelitev poslovnih zahtev (11)

Glob. model podatkovnih tokov


- Metoda dela:
 - Izdelava globalnega modela podatkovnih tokov lahko zajema:
 - Izdelavo kontekstnega diagrama - prikaže kontekst poslovnega sistema organizacije: medsebojno sodelovanje funkcionalnih področij ter sodelovanje sistema z okoljem.
 - Izdelavo področnih diagramov podatkovnih tokov, ki za posamezna funkcionalna področja prikažejo glavne procese ter podatkovne tokove.
 - Lahko se odločimo tudi za prikaz preko razširjenega kontekstnega vidika (primer na naslednji strani)

Opredelitev poslovnih zahtev (12)

Glob. model podatkovnih tokov



Opredelitev poslovnih zahtev (15)

Glob. model poslovnih procesov 

- Definicija:
 - Poslovni proces označuje množico povezanih aktivnosti, ki se izvajajo v organizaciji in posredno ali neposredno vplivajo na dodano vrednost pri uresničevanju skupnega cilja organizacije.
 - Globalni model poslovnih procesov je sestavljen iz grafične ter besedne predstavitve glavnih poslovnih procesov.
- Tehnika:
 - Poslovne procesa lahko predstavimo z različnimi tehnikami. Ena boljši je eEPC diagram.

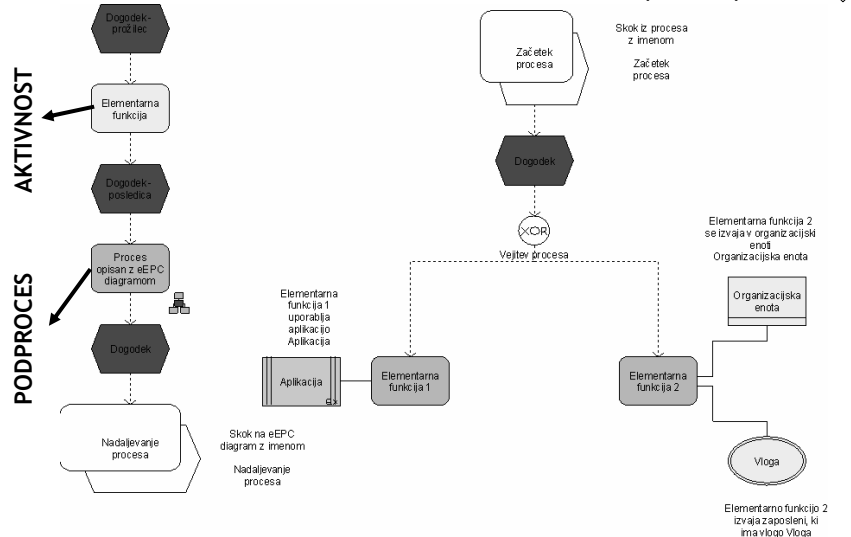
Opredelitev poslovnih zahtev (16)

Glob. model poslovnih procesov 

- Najpomembnejši gradniki eEPC tehnike so:
 - Aktivnost: aktivnost je končno zaporedje korakov oziroma operacij, ki ima neko opredeljeno časovno trajanje.
 - Dogodek: Dogodek je bodisi rezultat aktivnosti ali impulz, ki sproži aktivnost.
 - Krmilni tok: Krmilni tok prikazuje potek procesa.
 - Točke razvejitve in združevanja: Točke razvejitve in združevanja označujejo točke, kjer se kontrolni tokovi razvejijo oz. združijo. Za združevanje in razdruž. uporabljamo logične operatorje AND, OR in XOR.
 - Vloga: Vloga predstavlja subjekt, ki aktivnost izvaja oz. je zanj odgovoren (posameznik, skupina ljudi, organizacijska enota, ipd.)

Opredelitev poslovnih zahtev (17)

Glob. model poslovnih procesov



Opredelitev poslovnih zahtev (18)

Glob. model poslovnih procesov

- Analiza poslovnih procesov
 - Namen analize je zajeti najpomembnejše procese, ki se izvajajo v organizaciji ter identificirati morebitne nove procese za izboljšavo poslovanja.
 - Podrobnejša predstavitev in modeliranje poslovnih procesov je stvar projektov, ki sledijo. Ti za osnovo jemljejo izhodišča strateškega plana.
 - Analiza poslovnih procesov večkrat razkrije težave, ki nastanejo zaradi poteka procesov čez več funkcionalno ločenih enot.
 - Pojavlja se vprašanje, kdo je v določenem trenutku odgovoren za izvajanje poslovnega procesa.

Opredelitev poslovnih zahtev (19)

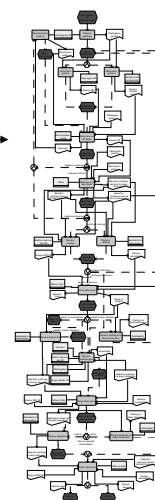
Glob. model poslovnih procesov

- Metoda dela:
 - V okviru strateškega plana zajamemo zgolj najpomembnejše poslovne procese. V tipični organizaciji je okrog 8 do 15 takšnih procesov.
 - Zajete procese prikažemo grafično (npr. Z uporabo eEPC diagramске tehnike) ter besedno opišemo.
 - Na voljo so številna orodja, ki omogočajo modeliranje poslovnih procesov v eEPC tehniki.
 - ARIS omogoča poleg modeliranja tudi simulacijo ter analizo učinkovitosti poslovnih procesov.
 - Viri za zajem glavnih poslovnih procesov so:
 - Obstoječa dokumentacija in
 - Delovni sestanki

Opredelitev poslovnih zahtev (20)

Glob. model poslovnih procesov

- PRIMER glavnih postopkov v telekomunikacijskem podjetju
 - Razvoj in vzdrževanje TK infrastrukture in sistemov
 - Dobava TK storitev
 - Zaračunavanje TK storitev
 - Zagotavljanje TK storitev
 - Trženje in razvoj TK storitev
 - Financiranje poslovanja
 - Nabava in logistika
 - Splošna podpora



Opredelitev poslovnih zahtev (21)

Glob. podatkovni model 

- Definicija:
 - Globalni podatkovni model je konceptualni model, ki prikazuje najpomembnejše koncepte organizacije ter povezave med njimi. Sestavljajo ga grafični prikaz ter opis entitet ali konceptov.
- Tehnika:
 - Za izdelavo globalnega podatkovnega modela uporabljamo diagram entiteta-razmerje. Atributov v sklopu strateškega planiranja ne zajemamo.

Opredelitev poslovnih zahtev (22)

Glob. podatkovni model 

- Konceptualno modeliranje
 - je tehnika, s katero skušamo na enostaven in razumljiv način predstaviti znanje, ki ga imamo o obravnavanem področju.
 - Z opazovanjem okolja, ki ga želimo modelirati, se dokopljemo do objektov in konceptov. Pri tem uporabljamo abstrakcijo - osredotočimo se na bistvene karakteristike, ostale pa zanemarimo.
 - Pri izdelavi poslovnega modela organizacije uporabimo konceptualno modeliranje za prikaz vseh pomembnih konceptov poslovnega okolja ter povezav med njimi.
 - Za risanje konceptualnih modelov uporabljamo različne tehnike. Med najbolj znane sodita tehnika entiteta-razmerje in razredni diagram.

Opredelitev poslovnih zahtev (31)

Povezovalne matrike 

- Povezovalne matrike so semantično bogata predstavitevna tehnika, uporabna za prikaz povezav med različnimi elementi, obravnavanimi v sklopu strateškega planiranja.
- V sklopu aktivnosti strateškega planiranja so zanimive predvsem matrike, ki povezujejo naslednje elemente:
 - Organizacijska enota,
 - Lokacija
 - Odgovorna oseba,
 - Funkcija,
 - Entiteta,
 - Poslovni proces,
 - Aplikacija,
 - Problem,
 - Cilj in
 - Tehnološka sprememba.

Opredelitev poslovnih zahtev (32)

Povezovalne matrike 

Povezovalne matrike	Organizacijska enota	Lokacija	Odgovorna oseba	Funkcija	Entiteta	Delovni proces	Aplikacija	Problem	Cilj	Tehnološka
Organizacijska enota										
Lokacija	x									
Odgovorna oseba										
Funkcija	x	x	x							
Entiteta	x			x						
Delovni proces				x	x					
Aplikacija				x	x					
Problem	x									
Cilj								x		
Tehnološka sprememba									x	

Opredelitev tehnoloških zahtev

Namen in koraki 

- Namen opredelitve tehnoloških zahtev je identificirati tiste potrebne tehnološke značilnosti sistema (kritični moduli sistema, distribuiranje podatkov in programov), ki bodo omogočale delovanje IS.
- Opravila zajemajo opredelitve tehnoloških zahtev zajemajo:

- Analizo vpliva IT
- Načrt IT

Analiza obstoječega stanja

Opredelitev poslovnih zahtev

Opredelitev tehnoloških zahtev

Planiranje IS

Dokumentacija

Opredelitev tehnoloških zahtev (2)

Analiza vpliva IT 

- Naloga vodstva je, da na organizacijo ne gleda samo tako kot je danes, ampak tako kakor bo v prihodnosti.
- Analiza informacijske tehnologije zajema obravnavo tehnoloških sprememb, ki kanejo vplivati na cilje in probleme organizacije oziroma njenega IS.
- Medsebojni vpliv problemov oziroma ciljev in tehnoloških sprememb lahko prikažemo s povezovalno matriko.

Opredelitev tehnoloških zahtev (3)

Analiza vpliva IT 

- Primeri tehnoloških sprememb:
 - računalniki: strežniki, delovne postaje, osebni računalniki (PC),
 - telekomunikacije,
 - internet/intranet/ekstranet,
 - arhitektura odjemalec/strežnik,
 - večnivojske arhitekture,
 - povezovalne tehnologije (CORBA, XML),
 - sistemi za upravljanje z znanjem,
 - sistemi za upravljanje z dokumenti,
 - poslovna inteligenca,
 - umetna inteligenca,
 - podatkovne baze,
 - sistemi za upravljanje PB in podatkovni center,...

Opredelitev tehnoloških zahtev (4)


Analiza vpliva IT 

- Elementi IT

Elektronsko poslovanje	
Objektna tehnologije	Portali (informacijski, storitveni)
	Povezovalne tehnologije (CORBA, XML, DCOM, EDI)
	Elementi IS (pod. baze, pod. skladišča...)
Arhitekture (odjemalec/strežnik, trinitvojska)	
Računalniška omrežja (LAN, WAN, intranet, ekstranet, internet)	
Strojna in komunikacijska oprema (računalniki, mrežna oprema...)	


Elektronsko poslovanje

Opredelitev tehnoloških zahtev (5)

Načrt IT 


- Načrt IT obsega specifikacijo predlaganih rešitev za:
 - strojno opremo,
 - komunikacijsko opremo in
 - programsko opremo.
- Načrt se lahko izdelava v več primernih različicah, ki ustrezajo danim omejitvam in zahtevam in med katerimi kasneje izberemo optimalno.

Opredelitev tehnoloških zahtev (6)

Načrt IT 


- Pri izdelavi načrta IT se držimo naslednje delitve IT:
 - Strojna računalniška oprema,
 - Komunikacijska oprema,
 - Programska oprema,
 - Kadri in
 - Drugo
- Vsako izmed skupin razdelimo na podrobne elemente. Kako podrobno gremo pri načrtu IT je odvisno predvsem od kompleksnosti IS.

Opredelitev tehnoloških zahtev (7)

Načrt IT 


- Strojna računalniška oprema
 - Standardna delovna postaja - osebni računalnik (PC)
 - Strežnik
 - Nestandardna delovna postaja
 - Prenosni računalnik
 - Docking postaja
 - Tiskalnik
 - Lokalni tiskalnik
 - Omrežni tiskalnik
 - Risalnik
 - Skener
 - Druga računalniška oprema

Opredelitev tehnoloških zahtev (8)

Načrt IT 


- Komunikacijska oprema
 - Stanje omrežja, ozka grla
 - Arhitektura, shema omrežja
 - Nadgraditev
 - Ostali gradniki omrežja (usmerjevalnik, ipd.)

Opredelitev tehnoloških zahtev (9)

Načrt IT 

- Programska oprema
 - Sistemska programska oprema
 - Operacijski sistem
 - Omrežni operacijski sistem
 - Razvojna programska oprema
 - SUPB
 - Sporočilni sistem
 - Uporabniški programi
 - Urejevalniki besedil
 - Preglednice
 - Predstavitveni programi
 - Slovarji
 - Elektronska pošta
 - Internet brskalnik
 - Aplikacije oziroma aplikativni sistemi
 - Specialne aplikacije
 - Skupne aplikacije

Opredelitev tehnoloških zahtev (10)

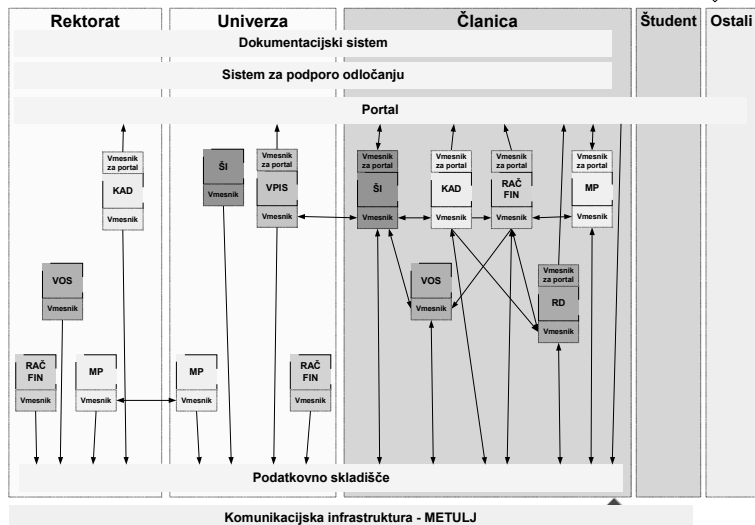
Načrt IT 

- Kadri
 - Izobraževanje
 - Organiziranost informatike v OS
- Drugo

Opredelitev tehnoloških zahtev (11)

Načrt IT

Primer načrta IT - predlog arhitekture aplikativnih sistemov

OSNOVE INFORMACIJSKIH SISTEMOV
2. Letnik VSP, Programska oprema

- 153 -

Opredelitev tehnoloških zahtev (12)

Načrt IT

Primer načrta IT - potrebna storjna oprema

Aplika tivni sistem	Instanca	Strežnik, Tip strežnika
VPIS	1	1, A
Kadrovski podsistem		
Raziskovalno delo	1	2, A
Vodenje osnovnih sredstev		
Študijska informatika	1	3, A
Materialno poslovanje		
Finančno-računovodski podsistem	1	4, B

Tip strežnika	Opis strežnika
A	1 procesor 2GB pomnilnika 2 diska
B	1 procesor 1GB pomnilnika 2 diska
C	1 procesor 1GB pomnilnika 2 diska polovične kapacitete
D	1 procesor 2GB pomnilnika 2 diska

OSNOVE INFORMACIJSKIH SISTEMOV
2. Letnik VSP, Programska oprema

- 154 -

Planiranje IS

Namen 

- Namen planiranja IS je izdelati terminsko in po sredstvih opredeljen načrt aktivnosti, ki so potrebne za uresničitev strateškega načrta.
- Aktivnosti so organizirane po naslednjih sklopih:
 - strojna računalniška oprema,
 - komunikacijska oprema,

V sklopu načrtovanja IT je bila osnovna naloga ugotoviti, kaj s področja predvsem informacijske tehnologije lahko vpliva na cilje in probleme organizacije oziroma IS ter določiti, kaj od strojne in programske opreme je potrebno zato nabaviti ter kako spremeniti organizacijo kadrov IS, da bodo le-ti služili namenu. Kdaj naj se nabava in spremembe izvedejo ter kolikšna sredstva so za to potrebna pa je stvar planiranja IS.

Planiranje IS (2)

Vsebina 

- V sklopu planiranja IS izvedemo naslednje korake, za katerimi so opravila:
 - Ocenimo strateški pomen aplikativnih sistemov
 - Določimo projekte za nadaljnji razvoj IS
 - Določimo prioritete posameznim projektom
 - Izdelamo plan aktivnosti po področjih
 - Strojna računalniška oprema
 - Komunikacijska oprema
 - Programska oprema
 - Kadri in drugo
 - Izdelamo plan sredstev in potreb po kadrih za vsako plansko leto posebej

Planiranje IS (2)

Tehnika 

- Tehnika:
 - Za večino izdelkov se pokažejo primerne matrike oziroma tabela, kjer po planskih letih prikažemo potrebe (po posameznih področjih) ter terminski plani (npr. antogram)
 - Primer: prioritete projektov

Področja – projekti	Prioriteta
Aplikativni projekti	
Kadrovska evidenca	1
Sistem plač za javne zavode	3
Postopek javnih naročil v DU	2
Kurir2	1, 2
Izvajanje zakona o družinskih prejemkih na CSD	1
Poškodbe pri delu	2

Planiranje IS (3)

Tehnika 

- Primer: matrika za prikaz stroškov vzdrževanja obstoječih aplikacij

Skupne aplikacije	Leto A	Leto B	Leto C
Infoklip			
ISPO			
IUS-INFO			
Specialne aplikacije	Leto A	Leto B	Leto C
Obresti			
Katalog podjetij			
Zaposlovanje			

Planiranje IS (4)

Tehnika 

- Primer: matrika za prikaz stroškov izobraževanja kadrov

OBDOBJE	KADRI (v delovnih mesecih)		SREDSTVA (v milijonih SIT)		
	Zunanji	Notranji*	Stroški		Skupaj
			Zunanji kadri	Notranji kadri	

Planiranje IS (5)

Tehnika 

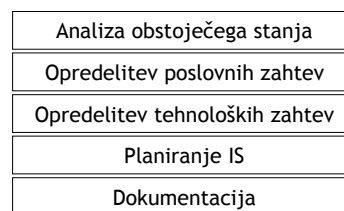
- Primer: matrika za prikaz stroškov zunanjih izvajalcev

	Računaln. Oprema	Komunik. Oprema	Program. oprema	Aplikativni projekti	Izobražev.	Drugo	Skupaj
1999							
2000							
2001							
SKUPAJ							

Izdelava dokumentacije

Namen 

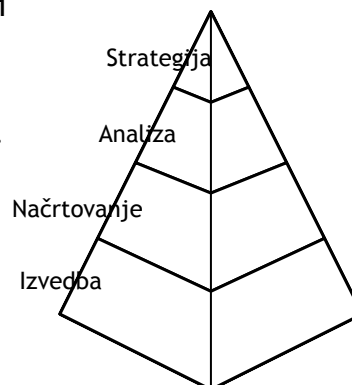
- Namen izdelave dokumentacije je predvsem izdelava slovarja pojmov, ki nastopajo kot izdelki ali pa se omenjajo v posameznih sklopi strateškega plana.
- Izdelava dokumentacije ni samostojna naloga temveč poteka vzporedno z drugimi aktivnostmi.



Analiza



Namen 

- Glavni namen analize je izdelati razumljiv opis realnega sveta oziroma poslovnega okolja, na katerega se nanaša razvoj IS.
- Analiza daje odgovor na vprašanje, KAJ naj IS podpira. Kaj se izvaja v poslovnih funkcijah in kakšne podatke te rabijo?



Analiza

Osnovni koraki 

- Analiza služi kot:
 - sredstvo za definicijo zahtev,
 - osnova za dogovor med naročnikom in izvajalcem
 - osnova za kasnejše faze razvoja.
- Osnovne aktivnosti analize zajemajo:
 - Zajem zahtev: zajem zahtev se nanaša na opredelitev funkcionalnosti, ki naj jo sistem podpira. Uporabniki sodelujejo z analitiki. 
 - Modeliranje sistema: predstavitev zajetih zahtev v razumljivi in nedvoumni obliki. Model analize navadno zajema več vidikov, ki so predstavljeni vsak z svojim modelom. 

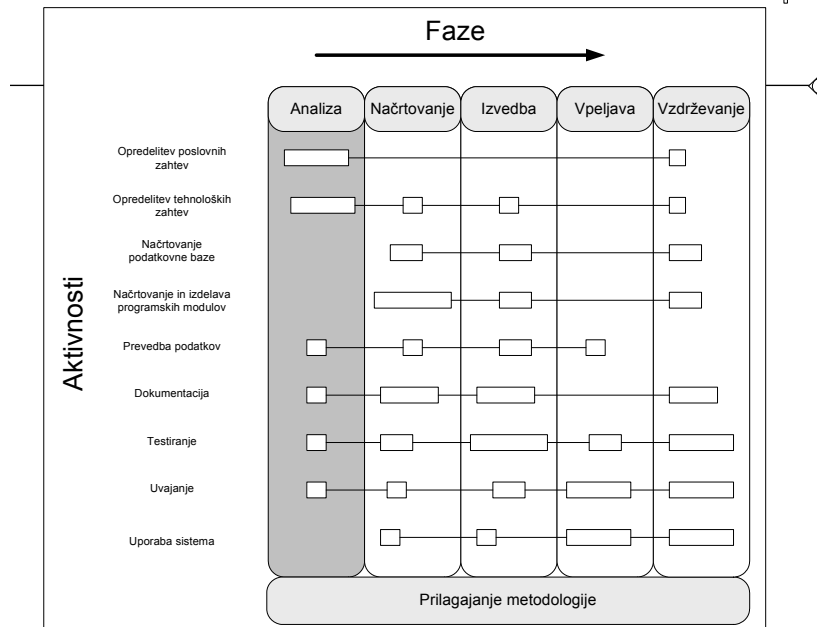
Analiza

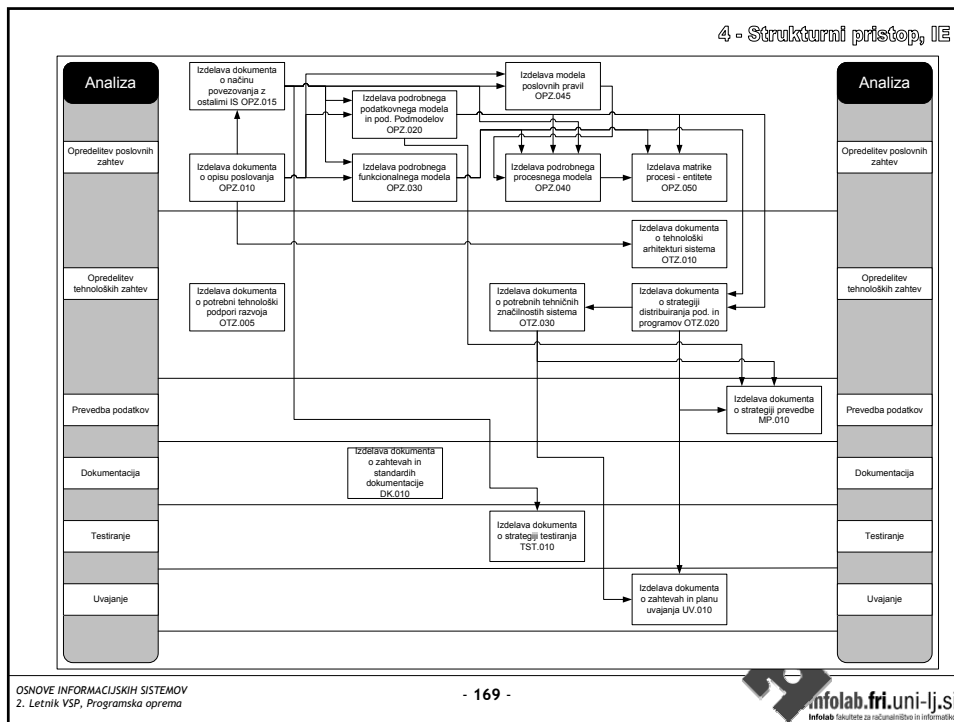
Ostale aktivnosti analize 

- Med ostale aktivnosti in opravila analize štejejo tudi:
 - Izdelava dokumenta o podpori sistemu
 - Izdelava dokumenta o strategiji distribuiranja podatkov in programov
 - Izdelava dokumenta o potrebnih tehničnih značilnostih sistema
 - Prevedba podatkov (AKTIVNOST)
 - Izdelava dokumenta o zahtevah in standardih dokumentacije
 - Izdelava strategije testiranja
 - Izdelava strategije uvajanja

Analiza

Povezava s strateškim planom





Analiza (4)

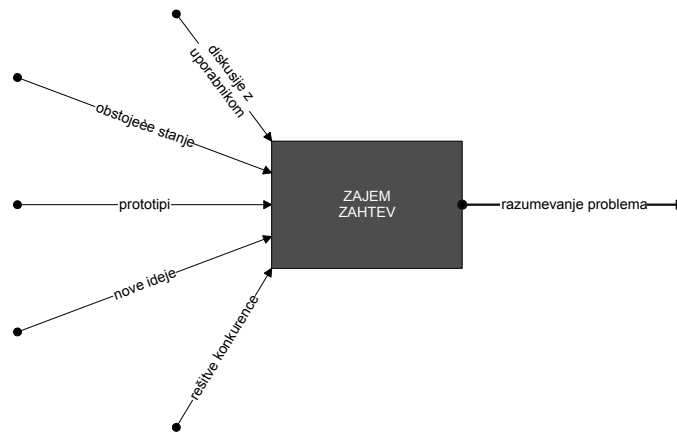
Zajem zahtev sistema

- Naloga zajema zahtev je zbrati čim več informacij o lastnostih obstoječega IS ter zahtev za nov IS.
 - Kako zaposleni izvajajo svoje delo?,
 - Kateri podatki so potrebni za pravilno delovanje sistema?,
 - Kakšne izpise je potrebno generirati?,
 - Kako uporabniki uporabljajo sistem pri svojem delu?,
 - So planirane spremembe načina dela v novem sistemu? ipd.
- Tipični elementi obravnave so:
 - Vhodi, postopki, izhodi, informacijski tokovi, podatkovne zbirke ipd.

Analiza (5)

Zajem zahtev sistema ◊

- Kaj je rezultat zajema zahtev?



Analiza (6)

Tehnike zajema zahtev ◊


- Zajem zahtev najlažje izvedemo z neposredno komunikacijo z udeleženci.
- Za zajem zahtev poznamo številne klasične in moderne tehnike zajema.
- Klasične tehnike
 - individualni in skupinski razgovori,
 - zajem informacij s pomočjo vprašalnikov,
 - opazovanje ljudi ob delu,
 - preučevanje obstoječe dokumentacije itd.

Analiza (7)

Tehnike zajema zahtev 

- Moderne tehnike:
 - uporaba CASE orodij,
 - uporaba prototipov itd.

Analiza (8)

Splošni napotki za zajem zahtev 

- Splošni napotki za uspešno izvedbo zajema zahtev:
 - Analitik mora biti objektivni,
 - Analitik mora upoštevati vse možnosti v okviru nekega problema,
 - Analitik posveča pozornost podrobnostim,
 - Analitik mora strmeti k novim in boljšim rešitvam,
 - Analitik ne daje obljub uporabnikom,
 - Analitik nima zadržkov pri zajemanju zahtev.

Tehnike zajema zahtev

Razgovori ali intervjuji ◊

- Razgovori ali intervjuji predstavljajo osnovno metodo zajema zahtev, pri kateri neposredno komuniciramo z ljudmi, udeleženi v procesih, ki jih želimo podpreti.
- Priporočljivo se je držati naslednjih smernic:
 - Razgovor je potrebno organizirati,
 - Na razgovor se je potrebno pripraviti,
 - Če je možno, razgovor snemamo,
 - Izogibati se moramo vprašanjem, ki nakazujejo odgovor,
 - Zapiske ali posnetek razgovora čim prej uredimo.

Tehnike zajema zahtev (2)

Razgovori ali intervjuji ◊

- Vprašanja, ki jih mislimo zastaviti v razgovoru, je potrebno v naprej razmisliti. Ločimo odprta in zaprta vprašanja.
- Odprta vprašanja:
 - Uporabimo takrat, ko ne vemo natančno, kaj želimo izvedeti od intervjujanca.
 - Iz odgovorov skušamo odkriti področja, ki so pomembnejša in na katera se je smiselno osredotočiti.
 - Primer odprtega vprašanja:

»Kaj pričakujete od novega sistema?«

Tehnike zajema zahtev (3)


Razgovori ali intervjuji 

- Zaprta vprašanja:
 - Ne dopuščajo poljubnih odgovorov, možne odgovore definirajo.
 - Uporabljamo takrat, ko so ključna vprašanja že razčiščena in se osredotočamo na podrobnosti.
 - Uporabimo za “prebijanje ledu”
 - Primer:



Kaj vas pri obstoječem sistemu najbolj moti?

- odzivni časi
- težaven dostop do podatkov
- neprijaznost uporabniškega vmesnika


Tehnike zajema zahtev (4)

Razgovori ali intervjuji 

Odperta vprašanja:

- 
 - Z odgovori na odprta vprašanja lahko odkrijemo pomembne informacije, ki bi nam ostale skrite, če bi možne odgovore sami definirali. Nove ugotovitve odpirajo nova vprašanja za nove raziskave.
 - Običajno pozitivno vplivajo na uporabnike, saj imajo občutek, da lahko odgovarjajo, kot sami želijo in jim nihče ne vsiljuje odgovorov.
- 
 - Odgovori na odprta vprašanja so lahko zelo dolgi in časovno potratni.

Tehnike zajema zahtev (5)

Razgovori ali intervjuji 

Zaprta vprašanja:



- Čas, potreben za odgovore na zaprta vprašanja, je relativno kratek. Uporaba zaprtih vprašanj zato omogoča, da postavimo več vprašanj in pokrijemo več področij.



- Ker zaprta vprašanja omejujejo možne odgovore, se lahko zgodi, da nam nekatere pomembne informacije ostanejo prikrite. Uporabnik se namreč trudi, da bi svoj odgovor našel v enem izmed definiranih odgovorov.

Tehnike zajema zahtev (6)

Vprašalniki 

- Vprašalniki so poleg razgovorov najpogostejša metoda zajema zahtev.
- Omogočajo vzporedno izvajanje - istočasno lahko vprašalnik izpolnjuje več ljudi - zato so časovno učinkoviti.
- Uporabljamo samostojno ali v kombinaciji z razgovori.
- Navadno ne anketiramo vseh uporabnikov, temveč izberemo le skupino. Pomembno je, da skupino ustrezno izberemo.
- Večinoma uporabljamo zaprta vprašanja.

Tehnike zajema zahtev (7)

Vprašalniki ◯

- Napotki:
 - Za izpolnjevanje vprašalnikov izberemo reprezentativno skupino,
 - Vprašalnike skrbno pripravimo,
 - Izpolnjene vprašalnike preučimo kmalu po izvedbi zajema zahtev,
 - Vprašalnike uporabimo takrat, ko nam razgovori zaradi časovne ali cenovne zahtevnosti ne ustrezajo, ali takrat, ko želimo zajeti informacije o neki specifični stvari, recimo, zanima nas splošno mnenje o nečem itd.

Tehnike zajema zahtev (8)

Opazovanje uporabnika pri izvajanju dela ◯

- Kadar se zdi, da uporabnik ne zna jasno povedati, kako opravlja svoje delo, uporabimo metodo opazovanja ob delu.
 - (+) pridobimo točne in realne informacije o delu, ki ga uporabnik opravlja,
 - (-) uporabnik se (morda) počuti nelagodno,
 - (-) uporabnik ne opravlja dela enako kot bi ga, če bi ga ne opazovali,
 - (-) čas opazovanja je relativno kratek - zakrite lahko ostanejo pomembne podrobnosti,
 - (-) časovna potratnost.

Tehnike zajema zahtev ⁽⁹⁾

Analiza obstoječega sistema ◊

- Analiza obstoječega sistema je komplementarna ostalim metodam zajema zahtev.
- Preučujemo dokumentacijo, ki se nanaša na:
 - Model obstoječega sistema,
 - Poslovni načrt,
 - Poslovna pravila,
 - Poročila,
 - Standardi,
 - Aktivnosti,
 - Delovne naloge ipd.

V pomoč so nam izdelki analize obstoječega stanja IS, ki smo jo izvedli v fazi strateškega planiranja

Tehnike zajema zahtev ⁽¹⁰⁾

Analiza obstoječega sistema ◊

- S proučitvijo dokumentacije lahko odkrijemo informacije, kot so:
 - problemi obstoječega sistema,
 - možnosti za izboljšave,
 - organizacijske usmeritve, ki lahko vplivajo na zahteve želenega sistema,
 - imena ključnih uporabnikov,
 - razlogi, zakaj je trenutni sistem tak, kot je,
 - podatki in storitve, ki so nujne za pravilno delovanje sistema,
 - pravila in principi izvajanja dela v organizaciji itd.

Moderne tehnike zajema zahtev

- Med moderne tehnike zajema zahtev uvrščamo:
 - uporaba sistemov za skupinsko delo (ang. Group Support Systems),
 - uporaba CASE orodij,
 - uporaba prototipov itd.

Moderne tehnike zajema zahtev (2)

Sistemi za skupinsko delo, CASE orodja, prototipi

- CASE orodja
 - Takojšnje modeliranje sistema
 - Diskutiranje na licu mesta
 - Zgodnje odkrivanje napak in pomanjkljivosti
 - Avtomatska izdelava prototipov
 - Navzkrižne kontrole
- Sistemi za skupinsko delo
 - Udeleženci enakopravno sodelujejo v razgovorih
 - Istočasno lahko sodeluje več udeležencev
 - Nobeden ne dominira, vsi lahko "govorijo" hkrati
 - Zagotovljena anonimnost

Modeliranje sistema

Splošno o modeliranju 

- Modeliranje je uveljavljena inženirska tehnika na mnogih področjih:
 - Gradbeništvo,
 - Avio-industrija,
 - Ekonomija,
 - Sociologija,
 - ...,
 - Računalniška programska oprema.

Modeliranje sistema (2)

Splošno o modeliranju 

- Model je poenostavitev realnosti, pri čemer je abstrakcija realnosti poljubno natančna.
- Pomembno je, da model prikazuje pomembne elemente in izpušča tiste, ki nas ne zanimajo.
- Modeliranje prinaša naslednje bistvene prednosti:
 - Omogoča vizualizacijo sistema,
 - Prikazuje tako statične kot dinamične lastnosti sistema,
 - Predstavlja šablono za nadaljnjo gradnjo sistema,
 - Dokumentira sprejete odločitve.
- Modele razvijamo zato, da bi sisteme bolje razumeli.

Modeliranje sistema (3)

Splošno o modeliranju 

- Izbira modelov
 - Modeliranja sistema se lahko lotimo na različne načine.
 - Izbira modelov ima pri tem pomembno vlogo - določa, kako bomo pristopili k reševanju problema ter kako oblikovali rešitev.
 - Modeli morajo podpirati izražanje na različnih ravneh natančnosti.
 - Najboljši modeli so tesno povezani z realnostjo.
 - En sam model nikoli ni dovolj. Sistem je potrebno modelirati iz različnih vidikov. Najboljši pristop je izbira nekaj modelov, ki kar najboljše pokrijejo najpomembnejše vidike sistema.
 - Metodologije razvoja IS predlagajo različne modele.

Modeliranje sistema (4)

Splošno o modeliranju 

- V splošnem na področju modeliranja poznana dva pristopa:
 - modeliranje iz vidika postopka in
 - modeliranje iz vidika objekta.
- Tradicionalni pristop k razvoju programske opreme je osnovan na postopkovni perspektivi:
 - osnovni gradnik sistema je procedura ali funkcija.
 - pogled usmerja razvijalca, da se osredotoči na potek postopkov in njihovo razgradnjo na manjše dele.
 - V praksi je pristop zelo dobro preizkušen in se veliko uporablja.
 - Zagovorniki novejših pristopov mu očitajo neprilagodljivost na vhodne spremembe.
- Modernejši pristop k razvoju IS je objektno usmerjen pristop. Osnovni gradnik takega pristopa je objekt.

Tradicionalno modeliranje

Vidiki trad. modeliranja ◊

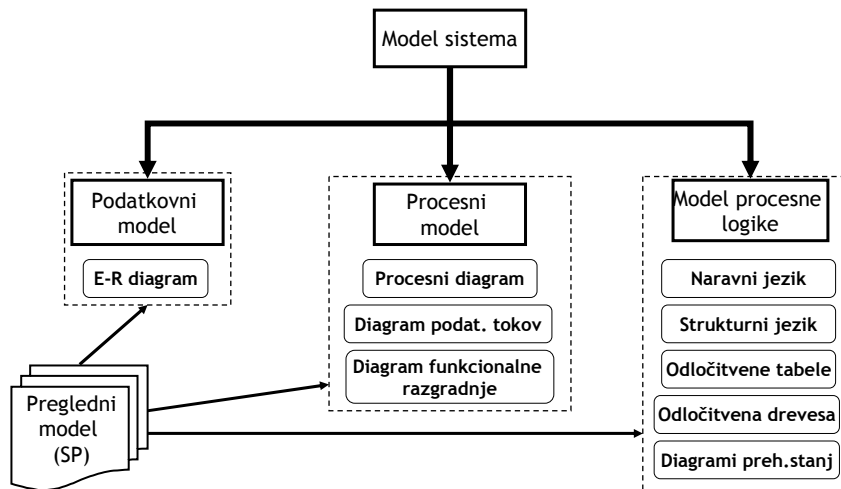

- Pri tradicionalnem modeliranju je specifikacija sistema sestavljena iz treh modelov, ki vsak s svojega vidika opisujejo sistem:
 - **Podatkovni model:** prikazuje sistem s podatkovnega vidika tako, da opisuje podatkovne strukture, ki so potrebne za delovanje sistema. Poleg podatkovnih struktur zajema tudi vse povezave med njimi.
 - **Procesni model:** prikazuje sistem z vidika aktivnosti ali procesov, ki se v sistemu izvajajo. Definirani so tokovi podatkov med procesi.
 - **Model procesne logike:** natančneje definira procese, definirane v procesnem modelu.

Tradicionalno modeliranje (2)

Tehnike trad. modeliranja ◊

- Za predstavitev posameznih modelov sistema uporabljamo formalne, semi-formalne in tudi neformalne tehnike.
 - Podatkovni model: diagram entiteta-razmerje
 - Procesni model: procesni diagram, diagram podatkovnih tokov, funkcionalna dekompozicija
 - Model procesne logike: naravni jezik, strukturiran jezik, odločitvene tabele, odločitveni grafi, diagrami prehajanja stanj

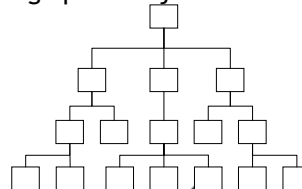
Tradicionalno modeliranje (3)

Povezava s SP 

Podatkovni model

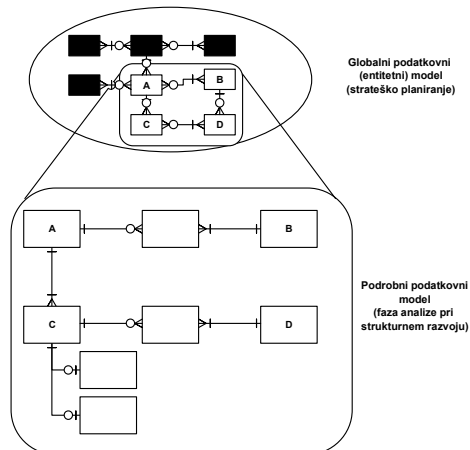
Definicija 

- Definicija:
 - Podatkovni model je eden izmed najpomembnejših izdelkov faze analize in predstavlja vse podatkovne kategorije, za katere na nekem delovnem področju obstaja potreba, da se o njih podatki spremljajo, obdelujejo in hranijo.
 - Vhod v podrobno podatkovno modeliranje je globalni podatkovni model - izdelek strateškega planiranja.



Podatkovni model (2)

Definicija




Podatkovni model (3)

Tehnika E-R

- Tehnika:
 - Za izdelavo podatkovnega modela uporabljamo diagrame entiteta-razmerje oz. entitetne diagrame.
- Osnove tehnike entiteta-razmerje:
 - Osnovni gradniki (ponovitev):
 - ENTITETA: realni ali abstraktni predmet obravnave, značilen za področje, o katerem zbiramo podatke: oseba, predmet, dogodek, pravilo, dejstvo, ..
 - RAZMERJE: Razmerje izraža pomensko povezavo med dvema entitetama in ima tudi atributiven značaj. Razmerje ima naslednje lastnosti:
 - ime: pove ime povezave,
 - vloga: pove vlogo entitete v povezavi,
 - števnost: pove koliko primerkov ene entitete nastopa v povezavi z enim primerkom druge entitete v povezavi.
 - Mandatornost: izraža, ali je entiteta obvezno v razmerju z drugo ali ne. Mandatornost lahko damo tudi pod okrilje števnosti

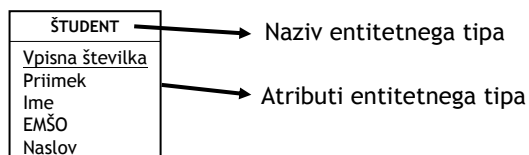

Podatkovni model (4)

Tehnika E-R 

(osnovni gradniki: nadaljevanje...)

- **ATRIBUT:** predstavlja lastnost entitete, tako da identificira, tipizira, poimenuje, opisuje ali kvalificira primerke entitet. Atributi se v grobem delijo na identifikacijske (osnovne), opisne (neosnovne) ter izpeljane atribute.
 - identifikacijski ali osnovni atribut: Z enoličnim identifikatorjem entitete se vsak primerek entitete med ostalimi primerki entitete enolično in nedvoumno identificira. Enolični identifikator entitete je lahko sestavljen iz enega ali več identifikacijskih atributov in razmerij.
 - opisni ali neosnovni atribut: opisni atributi opisujejo lastnosti entitete.
- **Posebna vrsta atributov so izpeljani atributi**
 - izpeljan atribut: vrednosti izpeljanih atributov se izračunajo iz definiranih vrednosti drugih atributov. Formule, algoritmi in logični izrazi za izračun vrednosti teh atributov so tudi del specifikacije podatkovnega modela.

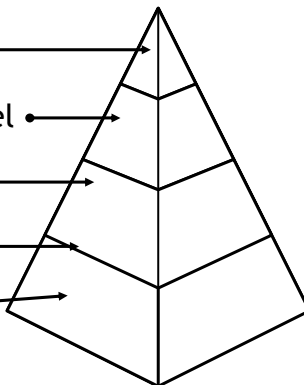
Podatkovni model (5)

Tehnika E-R 

Podatkovni model (6)

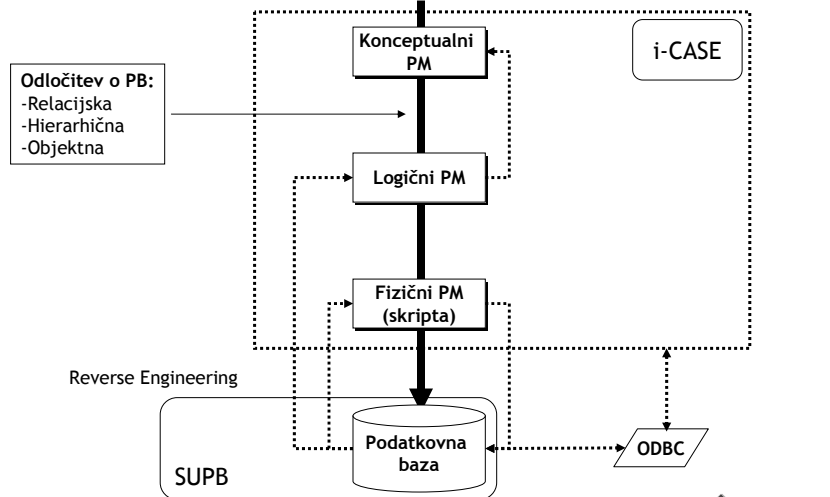
- Globalni podatkovni model
- Konceptualni podatkovni model
- Logični podatkovni model
- Fizični model
- Podatkovna baza

PM v fazi analize




Podatkovni model (6)

Od podatkovnega modela do PB



Podatkovni model (7)

Metoda dela 


- Metoda dela:
 - Postopki izgradnje diagrama entiteta-razmerje niso natančno predpisani. Za to obstajajo različne metode.
 - Globalni podatkovni model, ki nastane v sklopu SP je dobra osnova za izgradnjo podrobnega podatkovnega modela. Če globalni podatkovni model ne obstaja, je potrebno entitetne tipe in attribute identificirati drugače.
 - Izhajamo lahko tudi iz narave dejavnosti, za katero se gradi IS. S pomočjo vprašanj, ki jih zastavimo vodstvenim delavcem in ključnim uporabnikom, skušamo identificirati potrebne podatkovne strukture.

Podatkovni model (8)

Metoda dela 

- Eden od možnih pristopov za gradnjo podatkovnega modela:
 - Identificiramo entitetne tipe
 - Identificiramo lastnosti entitetnih tipov, ki nas zanimajo (atributi)
 - Vsak entitetni tip mora imeti enolični identifikator,
 - Pazljivi smo pri entitetah, ki nimajo identifikatorja iz vrst svojih atributov (odvisne entitete),
 - Izpeljani ali izvedeni atributi niso preveč zaželjeni,
 - Za attribute je potrebno smiselno izbirati podatkovne tipe ter jih čim več vključevati v domene.
 - Identificiramo razmerja oziroma povezave med entitetnimi tipi
 - Izogibamo so odvečnim povezavam
 - Vsaka povezava ima svoj pomen, entitetni tipi, ki jih povezuje, pa svojo vlogo v povezavi
 - Skrbno je potrebno preveriti števnost
 - Podatkovni model narišemo v več smiselno zaključenih delih, če se nam zdi, da bo zaradi velikega števila entitetnih tipov en sam diagram nepregleden.

Podatkovni model (9)

Metoda dela 

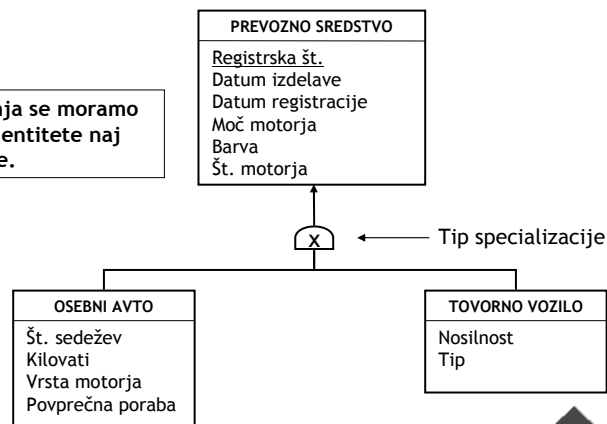
- Določanje podentitetnih tipov (specializacija ali generalizacija):
 - Osnovni in razširjeni diagram entiteta-razmerje (*EER-Extended entity-relationship diagram*).
 - Definicija podtipov neke entitete je odvisna od analitika, pri čemer velja slediti naslednjim pravilom. Podtipe je priporočljivo definirati v primerih, ko:
 - je definiranje podtipov logično in običajno,
 - obstaja atribut, ki je značilen samo za določeno podmnožico primerkov entitet določenega tipa,
 - obstajajo povezave podtipov entitet z drugimi entitetami ali podtipi entitet.

Podatkovni model (10)


Metoda dela 

- Primer specializacije/generalizacije

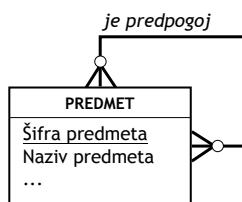
V fazi načrtovanja se moramo odločiti, katere entitete naj postanejo tabele.




Podatkovni model (11)

Metoda dela 

- Definiranje povezav - razmerij:
 - Povezava med dvema entitetnima tipoma izraža pomensko zvezo med entitetama.
 - Najpogostejše so povezave med dvema entitetama, možne pa so tudi rekurzivne povezave ter povezave med več entitetami (samo v nekaterih notacijah).
 - Pomembno je vedeti, kaj povzroči povezava v ciljnih podatkovni bazi!



Podatkovni model (12)

Metoda dela 

- Števnost povezav:
 - Pove, koliko primerkov enega entitetnega tipa nastopa v povezavi.
 - Za diagrame entiteta-razmerje obstajajo številne notacije. V orodju *Power Designer* se uporabljajo naslednji simboli:



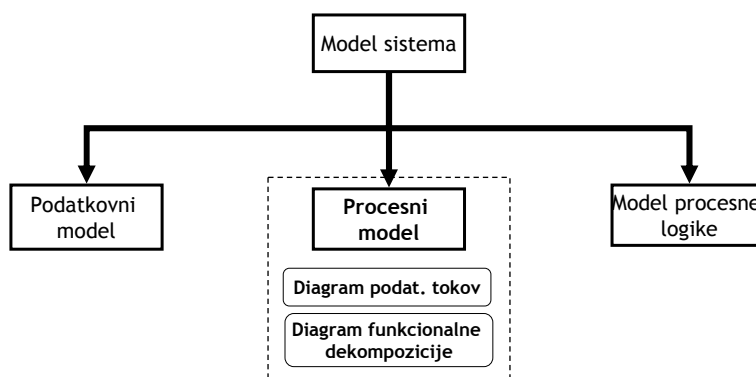
Podatkovni model (13)

Metoda dela ◊

- Opisovanje podatkovnega modela:
 - Ko smo podatkovni model grafično zaključili, moramo opisati entitetne tipe, atribute in povezave. Opis pripomore k večjemu razumevanju grafične predstavitve, pripomore pa tudi k odkrivanju nepravilnosti in pomanjkljivosti modela.
 - Opis naj zajema vsaj:
 - Opis entitetnih tipov: naziv in kratek opis,
 - Opis atributov za vsak entitetni tip posebej: naziv in kratek opis atributa, obveznost/neobveznost, domena, če obstaja,
 - Poleg entitetnih tipov in njihovih atributov naj besedni opis modela zajema tudi opis povezav med entitetami.

Procesni model

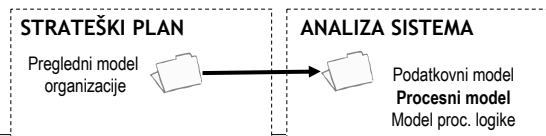
Umestitev ◊



Procesni model (2)

Definicija ◊

- Definicija:
 - Procesni model opredeljuje dinamično plat oz. vidik sistema. Prikazuje hierarhijo funkcij in procesov ter njihovo medsebojno odvisnost.
 - Za hierarhijo funkcij uporabljamo tudi izraz funkcionalni model
 - Vhod v procesno modeliranje so:
 - globalni funkcionalni model,
 - globalni model podatkovnih tokov in
 - globalni model poslovnih procesov.



Procesni model (3)

Tehnika ◊

- Tehnika:
 - Za predstavitev procesnega modela so nam na voljo naslednje tehnike:
 - diagram funkcionalne dekompozicije ali funkcionalna dekompozicija in
 - diagram podatkovnih tokov.
 - Omenjeni tehniki se uporabljajo tudi pri strateškem planiranju:
 - funkcionalna dekompozicija za globalni funkcionalni model,
 - diagram podatkovnih tokov za globalni model podatkovnih tokov.

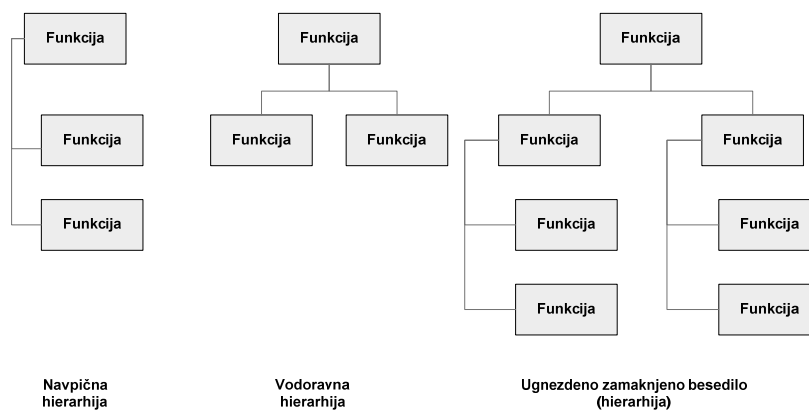
Procesni model (4)

Funkcionalna dekompozicija 

- Definicija:
 - Z diagramom funkcionalne dekompozicije ali krajše s funkcionalno dekompozicijo prikažemo hierarhijo funkcij, ki jih želimo:
 - s sistemom podpreti in sicer
 - Od tistih, ki se v sistemu dejansko izvajajo oz. iz katerih je v logičnem funkcijskem smislu sestavljen sistem
 - Osnova za risanje diagrama je globalni funkcionalni model, praviloma eno ali več njegovih funkcionalnih področij.
 - Hierarhijo funkcij lahko prikažemo na različne načine:
 - drevesna struktura kot navpična hierarhija pravokotnikov,
 - drevesna struktura kot vodoravna hierarhija pravokotnikov,
 - drevesna struktura kot ugnedeno zamaknjena (besedilo) hierarhija in
 - večnivojska drevesna struktura, kjer je pravokotnik vhod v strukturo na nižjem nivoju.

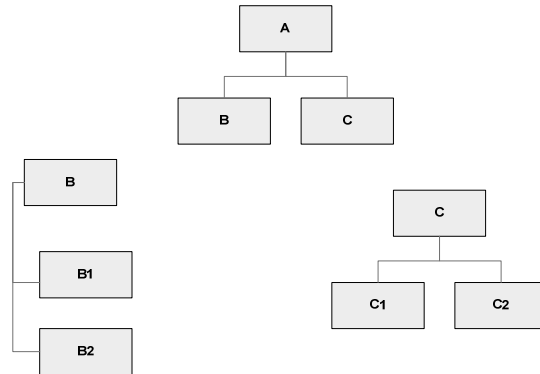
Procesni model (4a)

Funkcionalna dekompozicija 



Procesni model (4b)

Funkcionalna dekompozicija



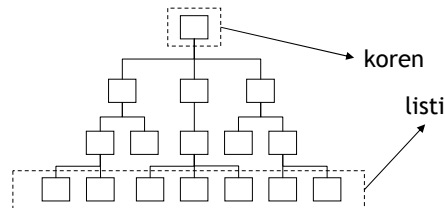
Večnivojska drevesna struktura

Procesni model (5)

Funkcionalna dekompozicija

- Značilnosti dekompozicije:
 - Vsaka hierarhična struktura se začne na vrhu z eno samo vseobsegajočo enoto - koren strukture (*root*).
 - Po najbolj enostavnem načinu prikaza so podrejene enote nanizane vodoravno en nivo nižje v hierarhiji in na podoben način do najnižjega nivoja. Elemente na najnižjem nivoju lahko zaradi podobnosti z drevesno strukturo imenujemo listi.

Vodoravna hierarhija



Procesni model (6)

Funkcionalna dekompozicija 

- Značilnosti dekompozicije (nadaljevanje):
 - Število nivojev in število enot na enem nivoju običajno ni omejeno. Velja priporočilo, naj ima vsak element največ devet (različna mnenja različnih avtorjev) podrejenih elementov.
 - Za vsako enoto velja, da ima lahko nič, eno ali več podrejenih enot (vej) in da vedno pripada natanko eni nadrejeni enoti na prvem višjem nivoju.
 - Enote na istem nivoju se razporedijo od leve proti desni po neki sekvenčni karakteristiki ali pa poljubno, pri čemer mora biti karakteristika natančno poznana in k diagramu dokumentirana.
 - Privzeto zaporedje opazovanja diagrama je sicer sicer od zgoraj navzdol in od leve na desno (skladno s sekvenčno karakteristiko).

Procesni model (7)

Funkcionalna dekompozicija 

- Metoda dela:
 - Gradnja podrobnega funkcionalnega modela poteka vzporedno z gradnjo podrobnega podatkovnega modela.
 - Prvi korak pri izdelavi podrobnega funkcionalnega modela je izdelava dekompozicijskega diagrama.
 - Drugi korak je opisovanje posameznih funkcij. Dve komponenti opisovanja funkcij:
 - Besedni opis
 - Za liste: Opis, katere entitete uporablja elementarna funkcija (list) in na kakšen način
 - Kje se ustavimo pri izgradnji modela je odvisno od CASE orodja in filozofije uporabe CASE orodja. Večina CASE orodij daje najboljše rezultate pri uporabi takrat, ko je elementarna funkcija (list) tista enota, ki jo je možno podpreti s formo ali izpisom

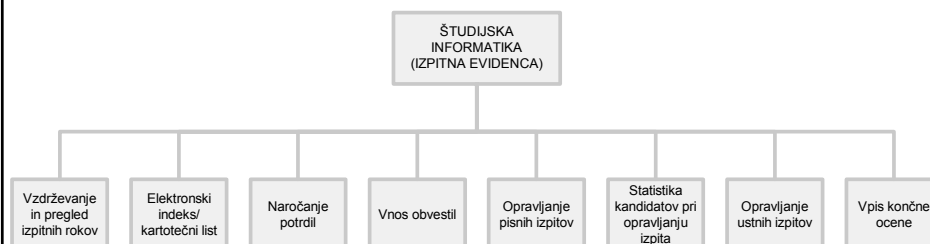
Procesni model (8)

Funkcionalna dekompozicija – primer ◊

- Primer: študijska informatika (izpitna evidenca)
 - Prvi nivo (funkcionalna področja):
 - Vzdrževanje in pregled izpitnih rokov,
 - Elektronski indeks/ kartotečni list
 - Naročanje potrdil
 - Vnos obvestil
 - Opravljanje pisnih izpitov
 - Statistika kandidatov pri opravljanju izpita
 - Opravljanje ustnih izpitov
 - Vnos končne ocene

Procesni model (9)

Funkcionalna dekompozicija – primer ◊



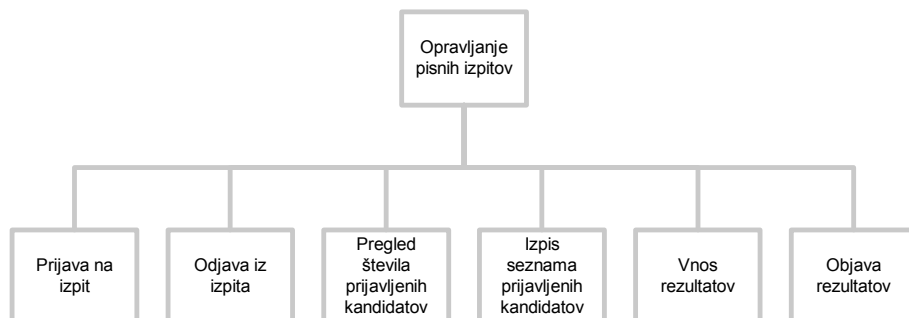
Procesni model (10)

Funkcionalna dekompozicija – primer ◊

- Primer (nadaljevanje):
 - Drugi nivo (za funkcijo opravljanje pisnih izpitov)
 - Prijava na izpit
 - Odjava iz izpita
 - Pregled števila prijavljenih kandidatov
 - Izpis seznama prijavljenih kandidatov
 - Vnos rezultatov
 - Objava rezultatov

Procesni model (11)

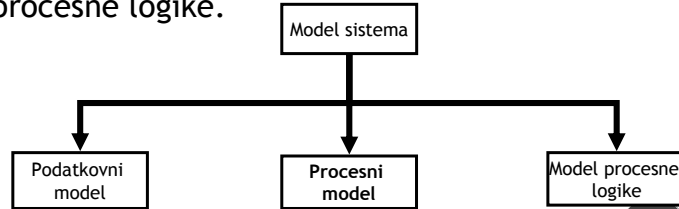
Funkcionalna dekompozicija – primer ◊



Procesni model (12)

Funkcionalna dekompozicija ◊

- Funkcionalna dekompozicija zajema:
 - Diagrame funkcionalne dekompozicije ter
 - opis funkcij
- V sklopu izdelave procesnega modela opišemo funkcije zgolj tekstualno (in preko uporabe entitet za liste). Formalni opisi so stvar modeliranja procesne logike.



Procesni model (13)

Diagram podatkovnih tokov ◊

- Definicija:
 - V fazi analize uporabimo diagrame podatkovnih tokov za prikaz povezanosti - sodelovanja z okoljem, v katerem bo sistem deloval ter za prikaz odvisnosti med procesi, ki jih bo sistem podprl.
- Tehnika:
 - Diagram podatkovnih tokov združuje podatkovni in procesni pogled na obravnavano področje.
 - Iz diagrama podatkovnih tokov je razvidna tudi hierarhija funkcionalne dekompozicije
 - Diagrame podatkovnih tokov je uvedel T. DeMarco leta 1978. Od takrat je nastalo več variant te tehnike. Razlikujejo se predvsem v notaciji.

Procesni model (14)

Diagram podatkovnih tokov 

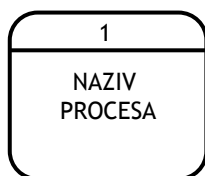
- Osnovni gradniki diagramov podatkovnih tokov:
 - Proces
 - Podatkovni tok
 - Podatkovno skladišče (shramba)
 - Zunanji izvor ali ponor (zunanja entiteta)
- Proces
 - Proces predstavlja v diagramu podatkovnih tokov množico aktivnosti, ki vhodne podatke pretvorijo v izhodne. Proces je generičen pojem za vse nivoje funkcionalnih komponent (poslovna funkcija, funkcija, elementarna funkcija, proces, podproces, naloga, ipd.)

Procesni model (15)

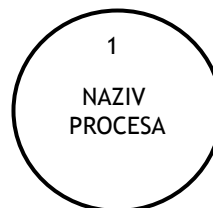
Diagram podatkovnih tokov 

- Grafični prikaz procesa

Gane-Sarson notacija



Yourdon-DeMarco notacija



Naziv procesa je glagol, glagolski samostalnik ali zaporedje besed, ki opisujejo vrsto dejavnosti. Poleg naziva procesa je procesu dodeljena številčna oznaka, ki proces enolično določa.

Procesni model (16)

Diagram podatkovnih tokov 

- Tok podatkov
 - Predstavlja množico vhodnih ali izhodnih podatkov, ki imajo enolično definirano vsebino in strukturo.
 - Naziv toka je samostalnik!
- Podatki, ki jih tok prikazuje, so lahko:
 - Elementarni podatek (ime, priimek, količina,...)
 - Dokument (račun, dobavnica, izpis iz rojstne matične knjige,...)
 - Množica dokumentov (projektna dokumentacija, Uradni list,...)

Procesni model (17)

Diagram podatkovnih tokov 

- Podatkovni tokovi lahko potekajo:
 - iz zunanjega izvora v proces ali iz procesa k zunanjemu ponoru,
 - iz procesa v drug proces in
 - iz procesa v skladišče podatkov ali obratno.

- Grafični prikaz toka podatkov

Grafični prikaz toka podatkov je v obeh notacijah (Gane-Sarson in Yourdon-DeMarco) enak.

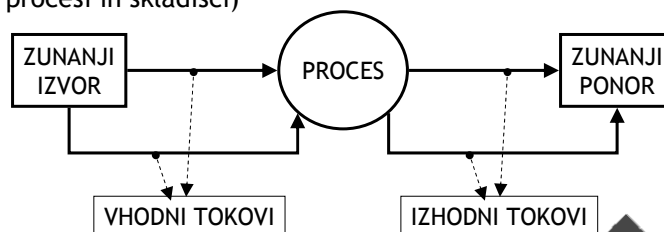
NAZIV TOKA PODATKOV
→

OPOMBA: naziv toka pove, kaj tok prenaša!! Za označevanje uporabljamo samostalnike v ednini ali pa kombinacijo samostalnika in pridevnika.

Procesni model (18)

Diagram podatkovnih tokov

- Glede na smer prenosa podatkov ločimo:
 - Vhodne tokove: potekajo od zunanjih izvorov do procesov, in
 - Izhodne tokove: potekajo od procesov do zunanjih ponorov podatkov.
 - Notranje tokove: potekajo interno (med procesi ali med procesi in skladišči)



Procesni model (19)

Diagram podatkovnih tokov

- Skladišče podatkov:
 - Je koncept, ki označuje prostor za shranjevanje podatkov iz nekega procesa, z namenom, da bodo ti na voljo tudi drugim procesom oziroma kasneje.
- Skladišče podatkov zajema različne medije za shranjevanje podatkov:
 - tabela, podatkovna baza, datoteka ipd.,
 - dokument,
 - mapa za shranjevanje dokumentov,
 - informacijsko-komunikacijski center (npr. knjižnica) ipd.

Procesni model (20)

Diagram podatkovnih tokov 

- V fazi analize se s skladiščem podatkov opisujejo logični sklopi podatkov. Ne zanima nas (še) njihova fizična organizacija.
- Grafični prikaz podatkovnega skladišča:

Gane-Sarson notacija



Yourdon-DeMarco notacija

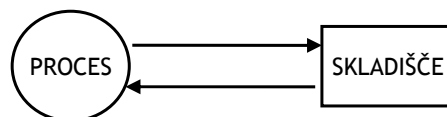


Naziv skladišča podatkov je največkrat enak nazivu vhodnih podatkovnih tokov. Skladišče je podatkovni tok v mirovanju. Skladišče ima podatkovno strukturo.

Procesni model (21)

Diagram podatkovnih tokov 

- Proces lahko opravlja dve vrsti operacij nad skladiščem:
 - Piše v skladišče (ažuriranje obstoječih podatkov, dodajanje in brisanje)
 - Bere iz skladišča
- Obstajajo tudi primeri, ko proces piše in bere iz istega skladišča.

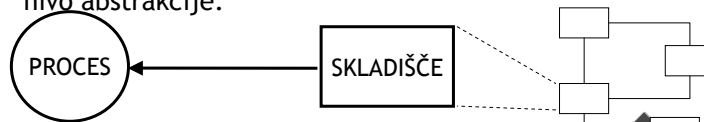


Proces piše in bere iz skladišča

Procesni model (22)

Diagram podatkovnih tokov 

- Povezava med podatkovnim in procesnim modelom:
 - Eden od načinov uporabe diagramov podatkovnih tokov je, da najprej izdelamo podatkovni model, potem pa z diagramom podatkovnih tokov pokažemo, kako se podatki med procesi pretakajo.
 - Skladišče podatkov tedaj ustrezajo entitetnim tipom iz podatkovnega modela. Vsebina in struktura skladišča je definirana s podatkovnim modelom.
 - Podatkovno skladišče lahko predstavlja tudi več entitet - nivo abstrakcije.



Procesni model (23)

Diagram podatkovnih tokov 

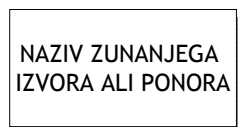
- Zunanji izvor ali ponor podatkov (zunanja entiteta):
 - Zunanji izvori ali ponori podatkov so koncepti, ki predstavljajo zunanje procese ali zunanje sisteme - subjekte.
 - Zunanji izvori in ponori podatkov se nahajajo izven interesnega področja naše analize, njihova struktura ali obnašanje nas ne zanimata.
 - Zanimajo pa nas podatkovni tokovi, ki jih povezujejo s prikazanimi procesi na diagramu podatkovnih tokov.
 - Nek zunanji sistem je lahko istočasno zunanji izvor in zunanji ponor tokov podatkov enega ali več procesov.

Procesni model (24)

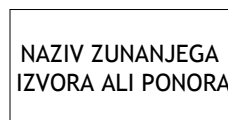
Diagram podatkovnih tokov 

- Grafična predstavitev zunanjega izvora ali ponora

Gane-Serson notacija



Yourdon-DeMarco notacija



Procesni model (25)

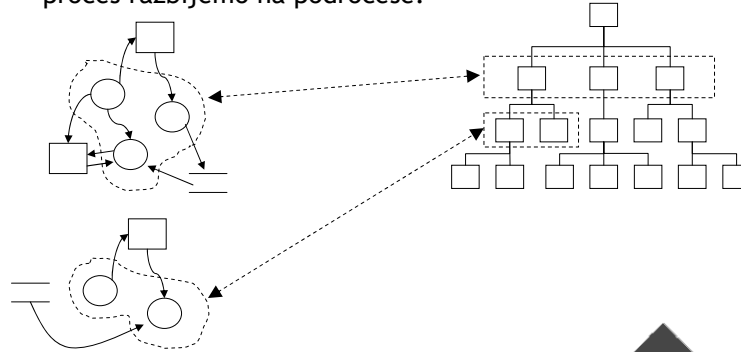
Diagram podatkovnih tokov 

- Razčlenjevanje diagramov podatkovnih tokov ali hierarhija diagramov podatkovnih tokov:
 - V analizi pogosto identificiramo večje število procesov (npr. nekaj sto).
 - Predstavitev vseh procesov enem diagramu je nepregledna, sama vsebina pa nerazumljiva.
 - Zato uporabljamo razčlenjevanje, s čimer diagrame rišemo od najvišjega nivoja, kjer nastopajo *obsežnejši* procesi, pa do najnižjega nivoja, kjer nastopajo *zelo podrobni* procesi.

Procesni model (26)

Diagram podatkovnih tokov

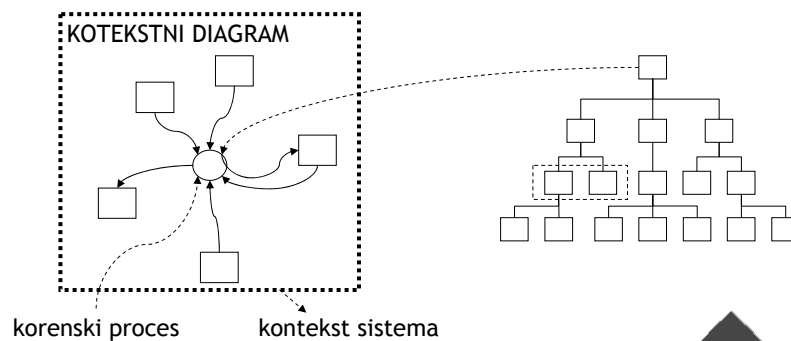
- Razčlenjevanje:
 - Za vsak proces, ki je predstavljen v diagramu na višjem nivoju, izdelamo poseben diagram podatkovnih tokov, kjer proces razbijemo na podprocese.



Procesni model (27)

Diagram podatkovnih tokov

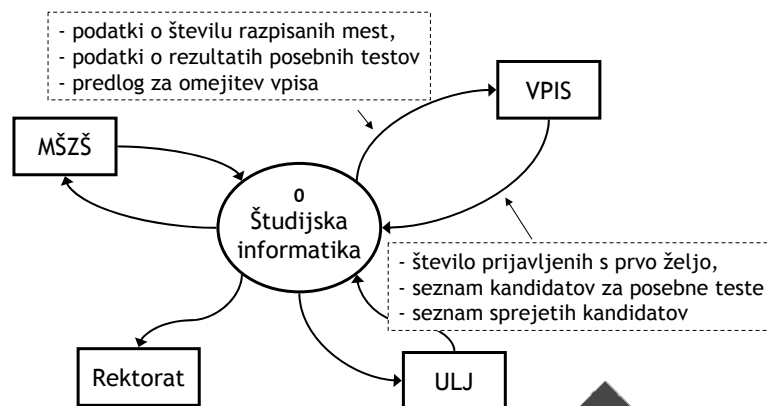
- Kontekstni diagram:
 - Razčlenjevanje diagramov podatkovnih tokov začnemo na najvišjem nivoju, kjer nastopa en sam proces - korenski proces.



Procesni model (28)

Diagram podatkovnih tokov

- Primer: kontekstni diagram sistema Študijska informatika



Procesni model (29)

Diagram podatkovnih tokov

- Značilnosti kontekstnega diagrama:
 - Kontekstni diagram prikazuje kontekst sistema - sistem v sodelovanju z okoljem
 - Kontekstni diagram ima en sam proces - korenski proces
 - Kontekstni diagram nima podatkovnih skladišč. Skladišča so namenjena odlagališču podatkov pri prenosu le-teh med procesi. Podatkovno skladišče je del sistema!
 - Podatkovni tokovi med korenskim procesom in zunanjimi entitetami opredeljujejo vmesnike med sistemom in okoljem.

Procesni model (30)

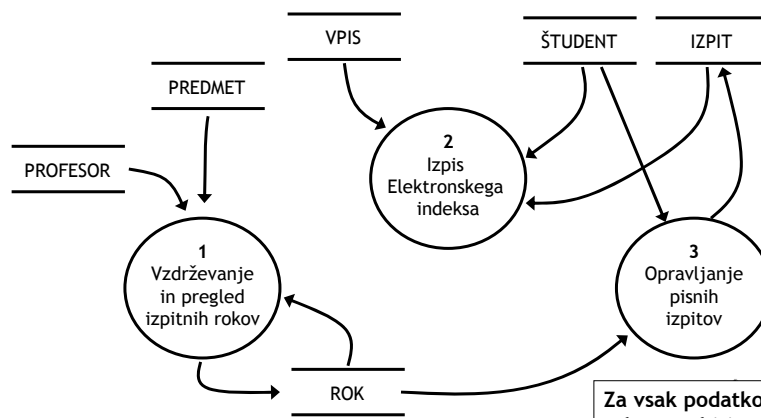
Diagram podatkovnih tokov

- Prvi nivo diagrama podatkovnih tokov
 - Prvi nivo razčlenitve kontekstnega diagrama predstavlja diagram podatkovnih tokov na hierarhičnem nivoju 1.
 - Diagram podatkovnih tokov na prvem hierarhičnem nivoju prikažemo z eno sliko, kjer je proces, predstavljen na kontekstnem diagramu, razčlenjen na potrebno število procesov (priporočljivo od 5 do 9).
 - Pri členjenju procesa je potrebno ohraniti vso funkcionalnost, kar pomeni, da je vsota funkcionalnosti vseh podrejenih procesov enaka funkcionalnosti nadrejenega procesa.
 - Potrebno je tudi zagotoviti, da so evidentirani procesi približno enakovredni oziroma uravnoteženi.

Procesni model (31)

Diagram podatkovnih tokov

- Primer: prvi nivo diagrama podatkovnih tokov sistema Študijska informatika (prikazan je samo del)



Za vsak podatkovni tok mora biti specficirano, kaj prenaša!!

Procesni model (32)

Diagram podatkovnih tokov 

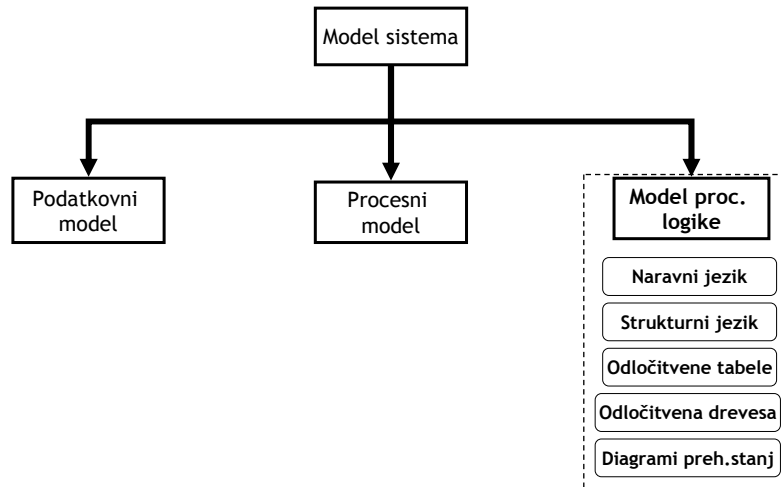
- Pravila risanja diagramov podatkovnih tokov
 - Procesi:
 - Kontekstni diagram ima en sam proces.
 - Procesi oz. njihovi diagrami, ki predstavljajo razčlenitev procesa na višjem nivoju, morajo zaobsegati vso njegovo funkcionalnost.
 - Na eni sliki (diagramu) ne prikazujemo več kot 9 procesov.
 - Podatkovna skladišča:
 - V vsako skladišče podatkov mora pisati vsaj en proces
 - Iz skladišča ni mogoče brati podatov, ki vanj niso bili zapisani.
 - Če skladišče podatkov uporablja en sam proces na nekem nivoju razčlenitve, potem je to skladišče odveč - spada na nižji nivo, kjer bo ta proces razčlenjen.

Procesni model (33)

Diagram podatkovnih tokov 

- Pravila risanja diagramov podatkovnih tokov (nadaljevanje)
 - Podatkovni tokovi:
 - Podatkovni tok mora izhajati ali prihajati v proces in ne more povezovati dveh skladišč podatkov.
 - Podatkovni tok ne more povezovati podatkovnega skladišča z zunanjo entiteto. Skladišče je del sistema, do katerega okolje nima neposrednega dostopa.
 - Vsako skladišče podatkov mora imeti vsaj en vhodni in en izhodni podatkovni tok, procesi, v katerih ti tokovi nastajajo ali se uporabljajo, pa se lahko nahajajo na drugih diagramih.

Model procesne logike




Model procesne logike (2)

Definicija

- Definicija:
 - Model procesne logike podrobno opisuje zaporedje korakov oziroma postopek pri izvedbi procesov, ki nastopajo na najnižji ravni v diagramu podatkovnih tokov ali v diagramu funkcionalne dekompozicije.
- Tehnike:
 - Naravni in strukturiran jezik
 - Odločitvene tabele
 - Odločitvena drevesa
 - Diagrami prehajanja stanj

Model procesne logike (3)

Logično in fizično modeliranje 

- Logično in fizično modeliranje procesne logike
 - Modeliranje procesne logike je del analize sistema.
 - Opisani procesi so logične enote, ki se v fazi načrtovanja lahko razčlenijo ali združijo v fizične programske module. Te je potrebno v fazi načrtovanja opisati s tehnikami, ki so povezane z izbranim razvojnim okoljem.
 - V nekaterih metodologijah ni razlike med konceptualnim in logičnem modeliranjem procesne logike.

Model procesne logike (4)

Formalne in neformalne metode 

- Formalne in neformalne tehnike:
 - Opisi postopkov morajo biti natančni, da lahko služijo kot osnova za nadaljnje načrtovanje oziroma kodiranje.
 - Po drugi strani morajo biti opisi tudi enostavni, da so lahko osnova za komunikacijo med analitiki, uporabniki in razvijalci.
 - Prednost formalnih opisov je jasnost in nedvoumnost, prednost neformalnih tehnik pa enostavnost in razumljivost.
 - V svetu obstajajo zagovorniki enih in drugih metod.

Model procesne logike (5)

Naravni jezik ◊

- Naravni jezik:
 - Naravni jezik je najpreprostejša oblika opisa nekega postopka.
 - Prednost naravnega jezika je njegova razumljivost - razume ga tudi netehnično osebje.
 - Slabost je dvoumnost - opisi dopuščajo različne interpretacije.

Model procesne logike (6)

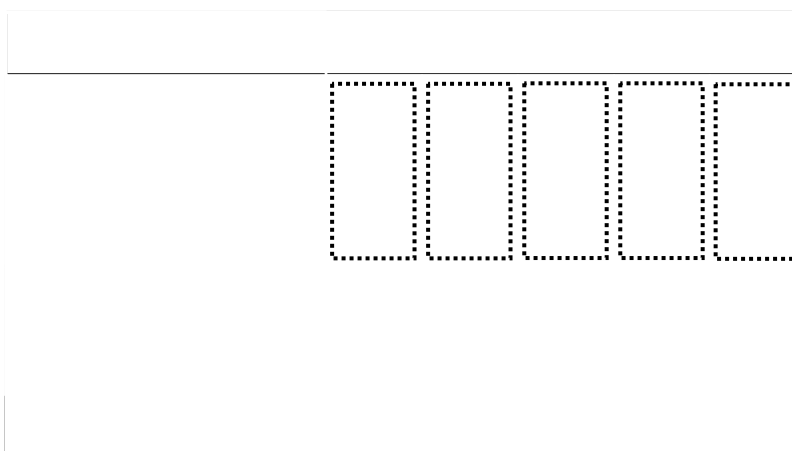
Strukturiran jezik ◊

- Strukturiran jezik:
 - Gre za izvedenko naravnega jezika. Opisi v strukturiranem jeziku so kratki in jedrnatih stavki, sestavljeni iz glagolskih in samostalniških oblik naravnega jezika.
 - Pri opisovanju v strukturiranem jeziku ne uporabljamo drugih besednih oblik, npr. pridevnikov, prislovov itd.
 - Strukturiran jezik pišemo z zamiki, da poudarimo strukturo posameznih delov opisa.

Model procesne logike

- Odločitvene tabele in drevesa:
 - Uporabljamo takrat, ko je v logiki procesa veliko pogojev, ki v različnih kombinacijah sprožajo različne akcije.
- Odločitvena tabela
 - Odločitvena tabela v zgornjem delu prikazuje pogoje, ki nastopajo v procesu ter vrednosti, ki jih ti pogoji lahko zavzamejo.
 - Posameznim kombinacijam vrednosti pogojev pravimo pravilo.
 - V spodnjem delu tabele so navedene akcije, ki se morajo izvesti ob določenem pravilu.

Model procesne logike



Model procesne logike

- PRIMER:

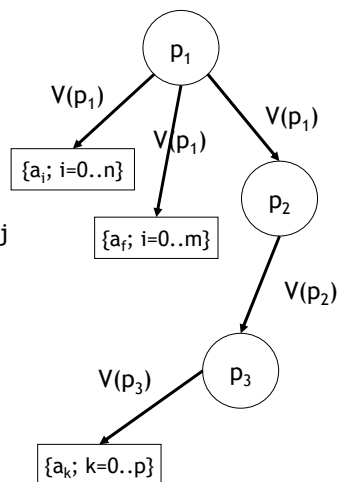
- odločitvena tabela za (poenostavljen) primer prijave na izpit

Pogoji akcije	Pravila							
Izpiti rok razpisan	D	N	D	N	D	N	D	N
Število dosedanjih opravljanj	<=3	<=3	>3	>3	<=3	<=3	>3	>3
Seminarska naloga opravljena	D	D	D	D	N	N	N	N
Sprejmi prijavo	x		x					
Zavrni prijavo		x		x	x	x	x	x
Sestavi komisijo			x					
Izdaj položnico za plačilo			x					

Model procesne logike

- Odločitveno drevo:

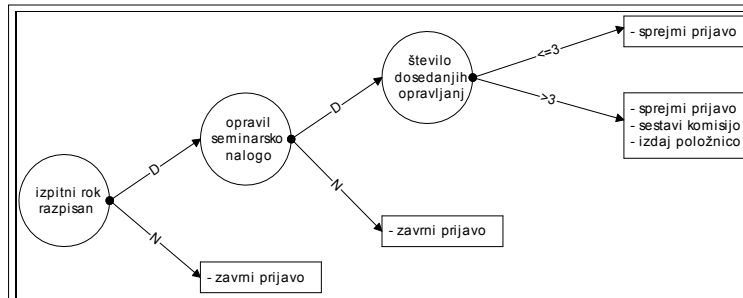
- Odločitveno drevo je sestavljeno iz vozlišč ter povezav med njimi.
- Vozlišča predstavljajo pogoje, povezave med njimi pa možne vrednosti posameznih pogojev.
- Iz vozlišča, ki predstavlja pogoj P1 (pogoj P1 lahko zavzame tri različne vrednosti; ZP1 = 3), gredo tri povezave.
- Posamezna pot v drevesu, od korena do predzadnjega vozlišča, predstavlja kombinacijo pogojev ali pravilo, list drevesa, ki je na koncu omenjene poti, pa prikazuje seznam akcij pravila.



Model procesne logike

- PRIMER:

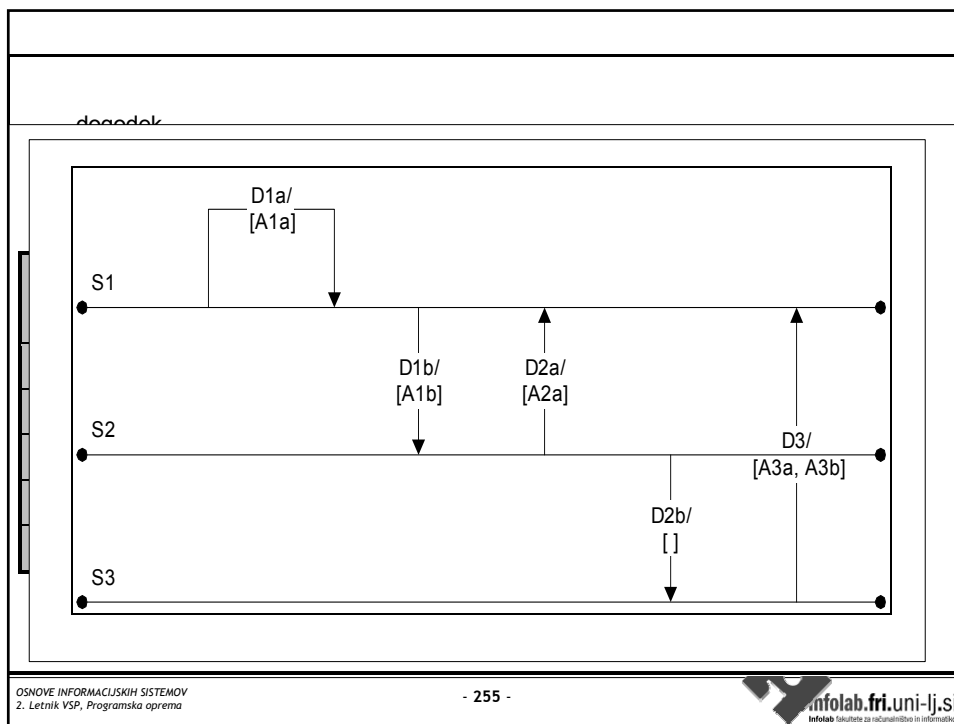
- odločitveno drevo za (poenostavljen) primer prijave na izpit



Model procesne logike


- Diagram prehajanja stanj:

- Omogoča prikaz:
 - Stanj, v katerih se sistem nahaja, in so za sistem pomembna,
 - Dogodkov, ki vplivajo na prehode med stanji,
 - Akcij, ki se zgodijo ob prehodih.
- Uporabimo vedno, ko imamo opravka s sistemi, ki se lahko nahajajo v različnih stanjih, od teh pa je odvisno, kako se odzivamo na različne dogodke.

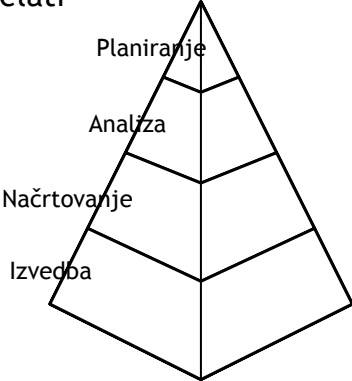


4 - Strukturni pristop, I/E

Načrtovanje

Namen 

- Glavni namen načrtovanja je izdelati načrt zgradbe sistema glede na specifikacije, ki so bile zbrane v fazi analize.
- Analiza daje odgovor na vprašanje, KAKO izdelati sistem, da bo ustrezal zahtevam, ki smo jih evidentirali v fazi analize.



OSNOVE INFORMACIJSKIH SISTEMOV
2. Letnik VSP, Programska oprema

- 256 -

infolab.fri.uni-lj.si
Infolab fakultete za računalništvo in informatiko

Načrtovanje (2)

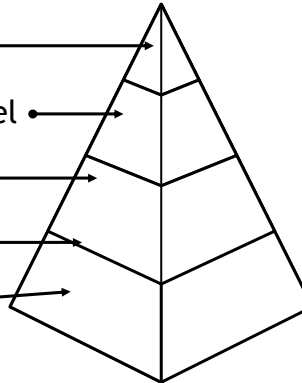
- Glavne cilji faze načrtovanja so:
 - izdelati načrt IS, ki ustreza specifikacijam, ki smo jih pridobili v fazi analize, in upošteva tehnološke omejitve sistema,
 - dokumentirati specifikacije načrta na način, ki bo omogočal vzdrževanje sistema in
 - zasnovati strategijo prehoda iz obstoječe na novo aplikacijo.

Načrtovanje (3)

- Aktivnosti, ki se odvijajo v fazi načrtovanja, so:
 - opredelitev tehnoloških zahtev,
 - načrtovanje podatkovne baze,
 - načrtovanje in izdelava programskih modulov,
 - prevedba podatkov,
 - dokumentacija,
 - testiranje,
 - uvajanje in
 - prehod na novi sistem.

Načrtovanje podatkovne baze

- Globalni podatkovni model
- Konceptualni podatkovni model
- Logični podatkovni model
- Fizični model
- Podatkovna baza



Konceptualni model je rezultat analize podatkov. V fazi načrtovanja ga pretvorimo v logični model ter nato v fizični model, ki je osnova za generiranje podatkovne baze.

Načrtovanje podatkovne baze (2)

- Kaj je podatkovna baza in kaj SUPB?
 - Podatkovna baza je zbirka med seboj pomensko povezanih podatkov, ki so shranjeni v računalniškem sistemu, dostop do njih je centraliziran in omogočen s pomočjo sistema za upravljanje podatkovnih baz
 - SUPB je sistem za upravljanje s podatkovnimi bazami:
 - zagotavljanje razpoložljivosti podatkov: učinkovit dostop vseh uporabnikov sočasno do vseh vrst podatkov ves čas
 - nadzor nad uporabo podatkov, v katerega okvir sodi skrb za:
 - celovitost (integriteto) podatkov, uporabo podatkov v skladu z njihovim namenom, uporabnost podatkov tudi v prihodnje

Načrtovanje podatkovne baze (3)

- Preden konceptualni model pretvorimo v logični model, moramo izbrati (vrsto) SUPB, v katerem bo naša podatkovna baza.
- Na voljo so razne vrste podatkovnih baz:
 - Relacijske baze
 - Objektne baze
 - Hierarhične baze
 - Mrežne baze

V veliki večini primerov se za potrebe poslovnih aplikacij uporablja relacijske podatkovne baze

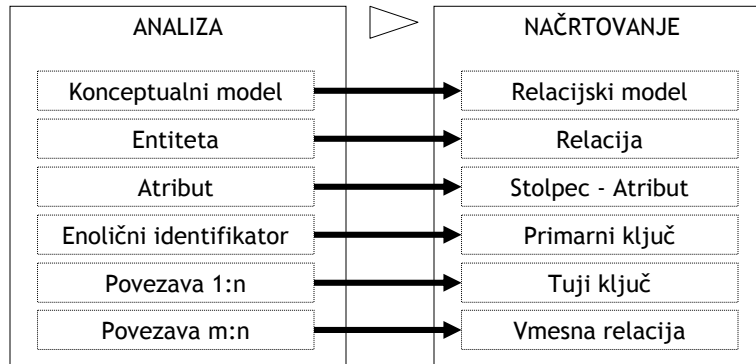
Načrtovanje podatkovne baze (4)

- Ob prehodu iz analize v načrtovanje se navadno odločimo kar za konkreten SUPB. Npr.:
 - Oracle,
 - MS SQL Server,
 - DB2,
 - InterBase,
 - Ingress ipd.
- CASE orodja večinoma podpirajo avtomatsko prehajanje iz faze analize v načrtovanje in obratno.

Načrtovanje podatkovne baze (5)

- Kaj se zgodi pri prehodu iz konceptualnega v logični model?

vrsta baze: relacijska
SUPB: Oracle



Načrtovanje podatkovne baze (6)

- V relacijskem modelu so podatkovne strukture predstavljene s pomočjo relacij.
 - Relacija je preslikava, katere domena je kartezijski produkt vrednostnih množic, območje pa dvojiška množica {res, ni res}

$$R: D_1 \times D_2 \times D_3 \times \dots \times D_n \rightarrow \{\text{res, ni res}\}$$

- Primer: če za nek izbor atributov, ki opisujejo osebo, velja, da res obstaja oseba s takimi lastnostmi, potem relacija tipa oseba preslika tak izbor atributov v resnično trditev.

Načrtovanje podatkovne baze (7)

- Relacijo lahko predstavimo z dvodimenzionalno tabelo, kjer stolpci predstavijo attribute relacije, vrstice pa podatke o entitetah.

$R(A_1, A_2, A_3, A_4, A_5, A_6)$ je relacijska shema relacije R

A1	A2	A3	A4	A5	A6
V(A1)	V(A2)	V(A3)	V(A4)	V(A5)	V(A6)
V(A1)	V(A2)	V(A3)	V(A4)	V(A5)	V(A6)
V(A1)	V(A2)	V(A3)	V(A4)	V(A5)	V(A6)

Načrtovanje podatkovne baze (8)

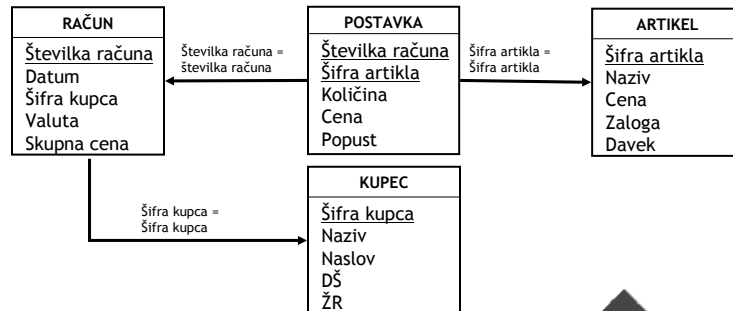
- Relacija je zaradi enostavnosti predstavljena s tabelo. Da je neka tabela relacija, mora veljati:
 - Vrednosti v stolpcih so atomarne, kar pomeni, da mora biti na presečišču vrstice in stolpca ena sama vrednost.
 - Vrednosti v nekem stolpcu pripadajo isti domeni, kar pomeni, da so vse vrednosti v nekem stolpcu zajete iz iste domene.
 - Vsaka vrstica je v tabeli enolična, kar pomeni, da v tabeli ne obstajata vrstici, ki bi bili popolnoma identični.
 - Zaporedje stolpcev v tabeli ni pomembno, kar pomeni, da je možno vrstni red stolpcev poljubno spreminjati, ne da bi s tem kakorkoli vplivali na pomen ali uporabnost relacije.
 - Tudi zaporedje vrstic v tabeli ni pomembno, kar pomeni, da je možno poljubno spreminjati tudi vrstni red vrstic.

Načrtovanje podatkovne baze (9)

- Primer:

Relacijska shema

Račun(Številka računa, Datum, #Šifra kupca, Valuta, Skupna cena)
 Postavka(#Številka računa, #Šifra artikla, količina, cena, popust)
 Artikel(Šifra artikla, naziv, cena, zaloga, davek)
 Kupec(Šifra kupca, naziv, naslov, DŠ, ŽR)



Načrtovanje podatkovne baze (10)

- Funkcionalne odvisnosti:

- Funkcionalna odvisnost je posebno razmerje med dvema atributoma oziroma skupinama atributov.
- V relaciji R velja, da je atribut B funkcionalno odvisen od atributa A, če je z vsako veljavno vrednostjo atributa A določena tudi vrednost atributa B.

$$A \rightarrow B$$

- Primer:

Oseba(EMŠO, PRIIMEK, IME, NASLOV,...)
 EMŠO → PRIIMEK

Načrtovanje podatkovne baze (11)

- Normalizacija relacijske podatkovne baze
 - Normalizacija je postopek pretvorbe kompleksnih podatkovnih struktur v enostavne, stabilne podatkovne strukture.
 - Nenormalizirane relacije so slabo strukturirane in obsegajo določeno stopnjo redundance, katere posledica so napake in nekonsistentnosti pri vnosu, spreminjanju in brisanju vrstic v tabeli.

Načrtovanje podatkovne baze (12)

- Postopek normalizacije
 - 1. Normalna oblika (1NO)**
 - Relacija je v prvi normalni obliki, kadar ne obstajajo ponavljajoče vrednosti atributov.
 - Relacijo je potrebno pretvoriti v prvo normalno obliko tako, da je v tabeli na vsakem presečišču vrstice in stolpca ena sama vrednost.

Načrtovanje podatkovne baze (13)

- 1NO - primer:

Študent(VŠ, priimek, ime, (šifra predmeta, ocena))

Atribut je sestavljen iz šifre predmeta in ocene.

VŠ	Priimek	Ime	Šifra predmeta, Ocena
64010632	Bratina	Simon	20020, 10
64016209	Telin	Tadeja	20060, 9
64016209	Telin	Tadeja	20061, 8

Načrtovanje podatkovne baze (14)

- Relacijo *Študent* normaliziramo v 1NO tako, da odpravimo ponavljajoče vrednosti atributov.

Študent(VŠ, priimek, ime, (šifra predmeta, ocena))

Študent(VŠ, priimek, ime)

Predmet(VŠ, šifra predmeta, ocena)

VŠ	Priimek	Ime	Šifra predmeta	Ocena
64010632	Bratina	Simon	20020	10
64016209	Telin	Tadeja	20060	9
64016209	Telin	Tadeja	20061	8

Načrtovanje podatkovne baze (15)

- Postopek normalizacije

2. Normalna oblika (2NO)

- Druga normalna oblika predpostavlja, da je relacija v prvi normalni obliki, hkrati pa ne smejo obstajati nepolne funkcionalne odvisnosti. To pomeni, da noben neosnovni atribut ne sme biti odvisen samo od podmnožice ključa.

Načrtovanje podatkovne baze (15)

- 2NO - primer:

Postavka(#šifra računa, #šifra artikla, naziv artikla, količina, cena)

Naziv artikla je odvisen samo od šifre artikla!

Šifra računa	Šifra artikla	Naziv artikla	Količina	Cena
1001	205A	Mehanizem za s24	2	2.990
1001	C890U	Safirno steklo	1	16.970

Načrtovanje podatkovne baze (16)

- Relacijo *Postavka* normaliziramo v 2NO tako, da jo razbijemo v dve relaciji.

Postavka(#šifra računa, #šifra artikla, naziv artikla, količina, cena)



R1(#šifra računa, #šifra artikla, količina, cena)

R2(šifra artikla, naziv artikla)

Šifra računa	Šifra artikla	Količina	Cena
1001	205A	2	2.990
1001	C890U	1	16.970

Šifra artikla	Naziv artikla
205A	Mehanizem za s24
C890U	Safirno steklo

Načrtovanje podatkovne baze (17)

- Postopek normalizacije
 - 3. Normalna oblika (3NO)**
 - Tretja normalna oblika predpostavlja, da je relacija v drugi normalni obliki, hkrati pa ne smejo obstajati neosnovni atributi, ki bi bili odvisni od drugega neosnovnega atributa (tranzitivna odvisnost).

Načrtovanje podatkovne baze (18)

- 3NO - primer:

Študent(VŠ, priimek, ime, poštna številka, kraj)

Funkcionalne odvisnosti:

VŠ → priimek, ime, poštna številka, kraj

Poštna številka → kraj

VŠ	Priimek	Ime	Poštna številka	Kraj
64010632	Bratina	Simon	1000	Ljubljana
64016209	Telin	Tadeja	5270	Ajdovščina
64010670	Berce	Marjan	4000	Kranj

Načrtovanje podatkovne baze (19)

- Relacijo *Študent* normaliziramo v 3NO tako, da jo razbijemo v dve relaciji.

Študent(VŠ, priimek, ime, poštna številka, kraj)



R1(VŠ, priimek, ime, #poštna številka)

R2(Poštna številka, kraj)

VŠ	Priimek	Ime	Poštna številka
64010632	Bratina	Simon	1000
64016209	Telin	Tadeja	5270
64010670	Berce	Marjan	4000

Poštna številka	Kraj
1000	Ljubljana
5270	Ajdovščina
4000	Kranj

Načrtovanje podatkovne baze (19)

- Postopek normalizacije

4. Poslovna normalna oblika (4NO)

- Relacija je v četrti poslovni normalni obliki, če je v tretji normalni obliki in ustreza enemu od naslednjih pogojev:
 - njeni atributi so odvisni ne samo od celotnega osnovnega ključa, ampak tudi od vrednosti tega ključa ali
 - je bil nek atribut premeščen iz relacije, kjer je bil opcijski v relacijo, kjer je v celoti odvisen od ključa in mora obstajati (je obvezen).

Načrtovanje podatkovne baze (20)

- 4PNO - primer:

Študent(VŠ, priimek, ime, datum plačila, status)

Datum plačila in status sta posebna atributa. Datum plačila je pomemben, ko gre za izrednega študenta, status pa za redno vpisane študente.

VŠ	Priimek	Ime	Datum plačila	Status
64010632	Bratina	Simon		Ponavljalec
64016209	Telin	Tadeja	19.4.2001	
64010670	Berce	Marjan	12.4.2001	
64620010	Mele	Silvana		Pavzer
65120987	Leban	Tibor		Prvi vpis

Načrtovanje podatkovne baze (21)

- Relacijo *Študent* normaliziramo v 4PNO tako, da jo razbijemo v tri relacije.

Študent(VŠ, priimek, ime, datum plačila, status)



R1(VŠ, priimek, ime)

R2(VŠ, datum plačila)

R3(VŠ, status)

VŠ	Priimek	Ime
64010632	Bratina	Simon
64620010	Mele	Silvana
65120987	Leban	Tibor

VŠ	Status
64010632	Ponavljalec
64620010	Pavzer
65120987	Prvi vpis

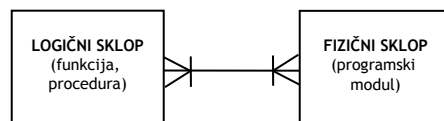
VŠ	Datum plačila
64016209	19.4.2001
64010670	12.4.2001

Načrtovanje podatkovne baze (12)

- V fazi načrtovanja podatkovne baze poskrbimo tudi za:
 - Indekse,
 - Poglede,
 - Sprožilce,
 - Bazne programe,
 - Podatkovno integriteto in
 - Omejitve.
- Gre za posege, ki jih omogoča razvoj s CASE orodjem. Vsebinsko ti posegi sodijo (tudi) v fazo izvedbe.

Načrt strukture modulov

- Funkcije in procesi, ki jih obravnavamo v fazi analize, predstavljajo logične sklope sistema. V fazi načrtovanja jih pretvorimo v fizične oz. programske sklope ali module.
- Implementacija enega logičnega sklopa je lahko izvedena z več programskimi sklopi. En programski sklop pa lahko implementira tudi več logičnih enot.



Načrt strukture modulov (2)

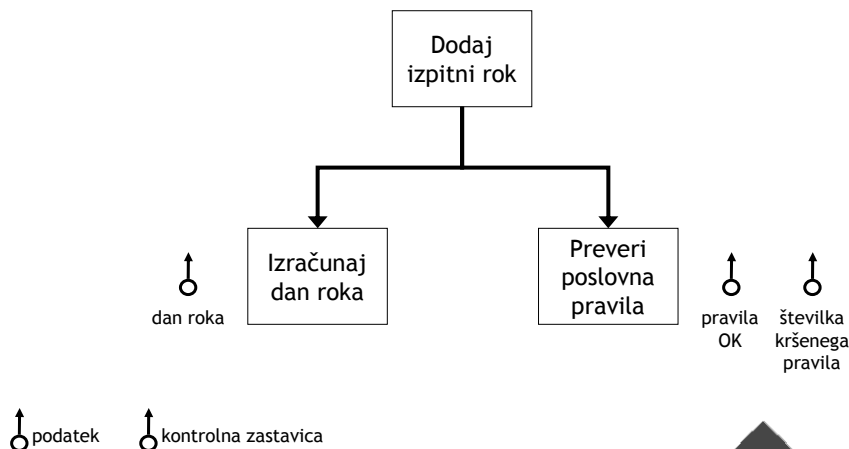
- Tehnika:
 - Strukturno programskih modulov prikažemo s pomočjo strukturnega diagrama.
- Lastnosti strukturnih diagramov:
 - Strukturni diagram prikazuje, kako je aplikativni sistem organiziran s hierarhijo programskih modulov.
 - Iz diagrama je razvidna odvisnost med moduli aplikativnega sistema z vidika podatkov, ki se med moduli prenašajo.
 - Iz strukturnega diagrama so razvidni tudi osnovni elementi strukturnega programiranja: zaporedje, izbira in ponavljanje.

Načrt strukture modulov (3)

- Lastnosti strukturnih diagramov (nadaljevanje):
 - Moduli so organizirani v hierarhijo, podobno kot funkcije v funkcionalni razgradnji.
 - Na najvišjem mestu je vseobsegajoč modul ali koren. Na naslednjem nivoju so moduli, ki jih koren lahko kliče (analogno kot izbire v meniju).
 - Moduli komunicirajo med seboj s pomočjo parametrov:
 - nosilci podatkov
 - kontrolne zastavice

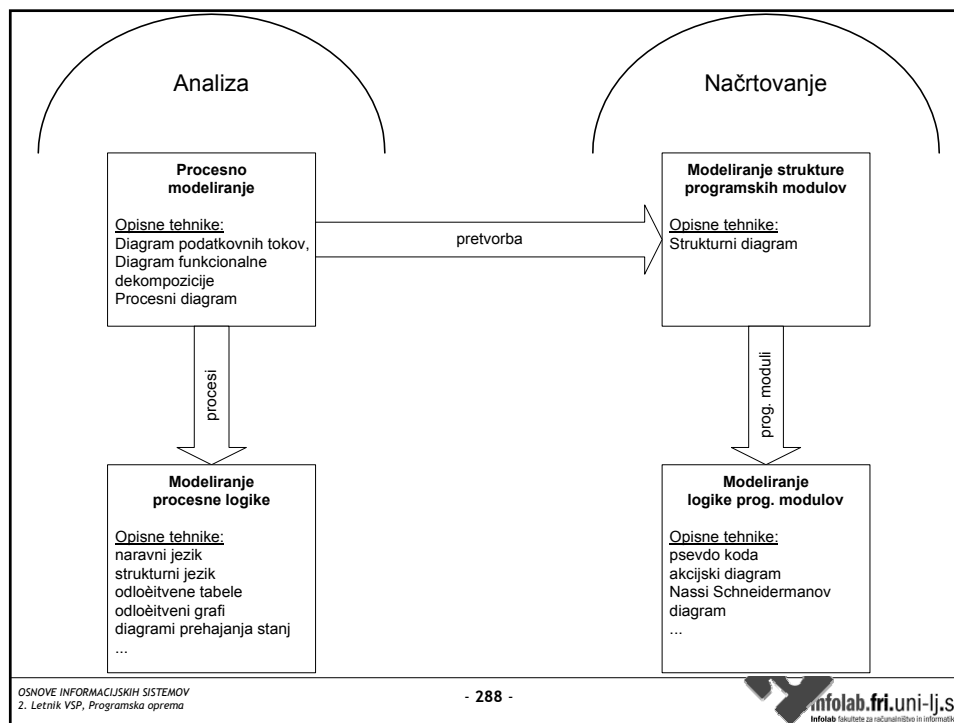
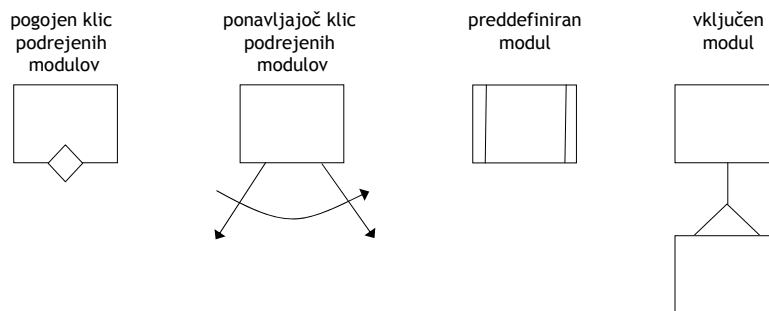
Načrt strukture modulov (4)

- Grafična notacija:



Načrt strukture modulov (5)

- V diagramski tehniki strukturnih diagramov se uporabljajo posebni grafični simboli, ki pomenijo:
 - določeno vrsto obdelave ali
 - določeno vrsto modula



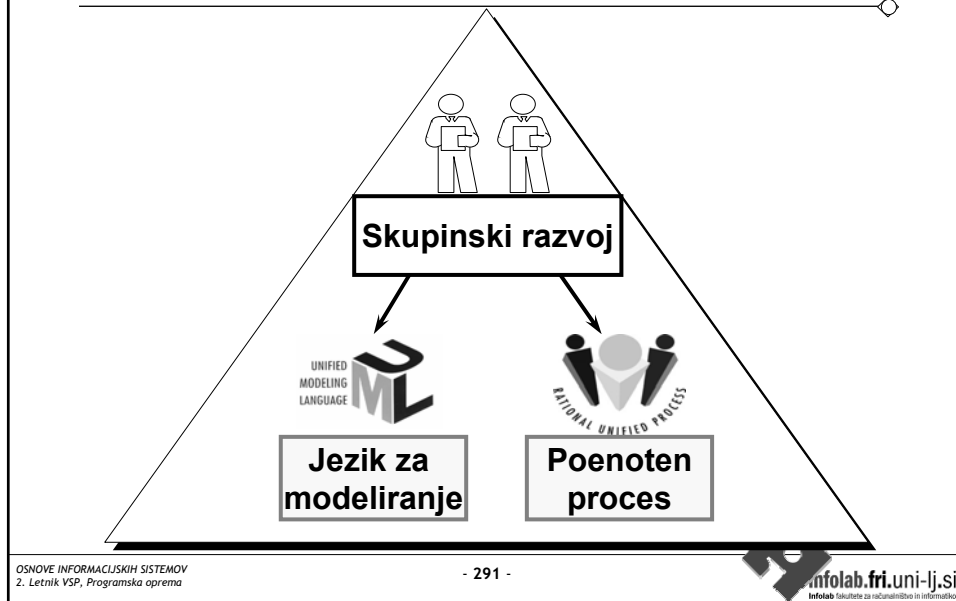
5 OBJEKTNI PRISTOP, MODELIRNI JEZIK *UML*- *Unified Modeling Language* in METODOLOGIJA *RUP-Rational Unified Process*

- Sklop 5.1: Modelirni jezik UML in proces razvoja RUP
- Sklop 5.2: Osnove objektne usmerjenosti
- Sklop 5.3: Podrobneje o diagramih UML

Sklop 5.1 - VSEBINA

- **MODELIRNI JEZIK UML IN PROCES RAZVOJA RUP**
 - O modelirnem jeziku UML
 - RUP - proces razvoja programske opreme

Pri gradnji sistema jezik ni dovolj

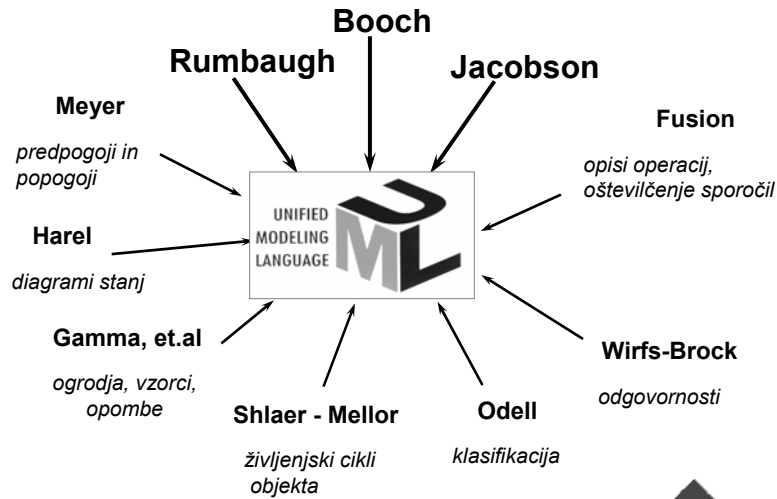


Kaj je UML?

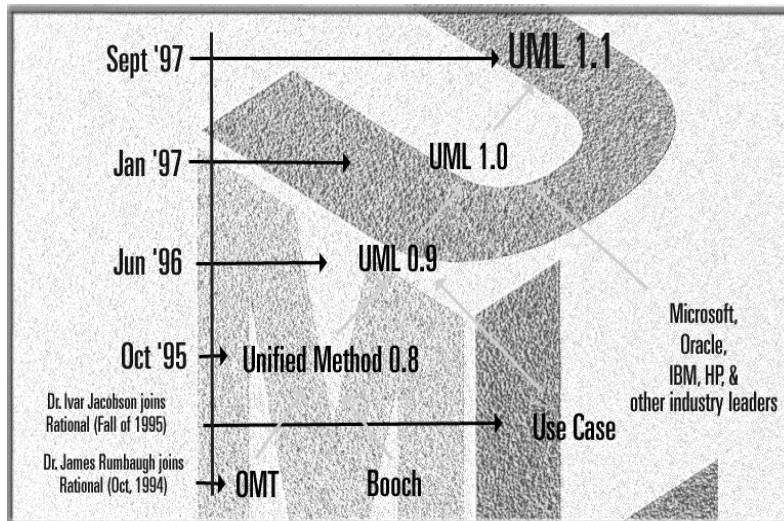
- Unified Modeling Language (UML) je jezik za
 - specifikacijo
 - vizualizacijo
 - konstrukcijo
 - dokumentacijo



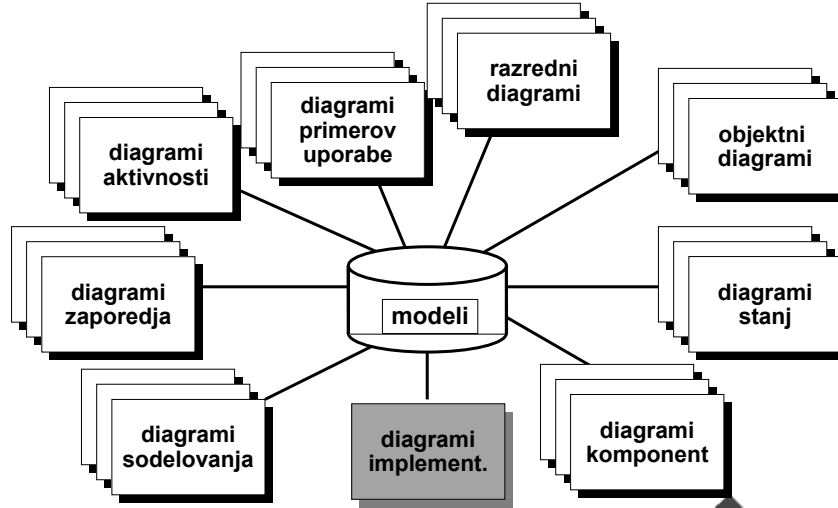
Viri, ki so prispevali k UML



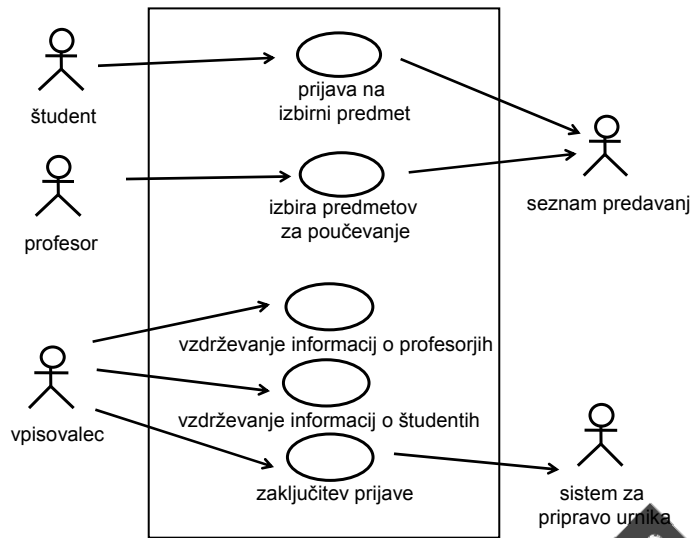
Zgodovina UML



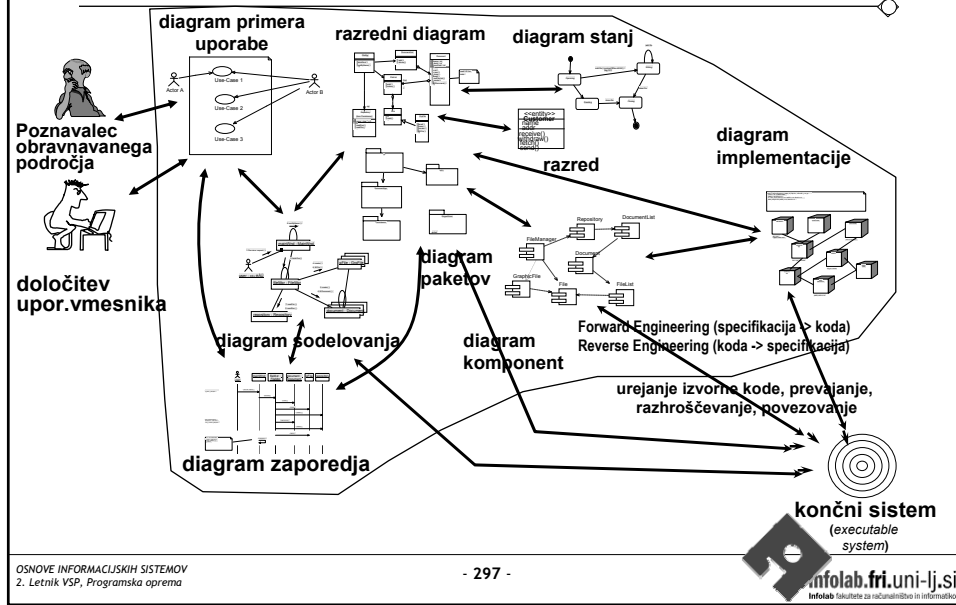
UML diagrami



Primer diagrama UML: Primer uporabe



Diagrami UML so hrbtenica razvoja



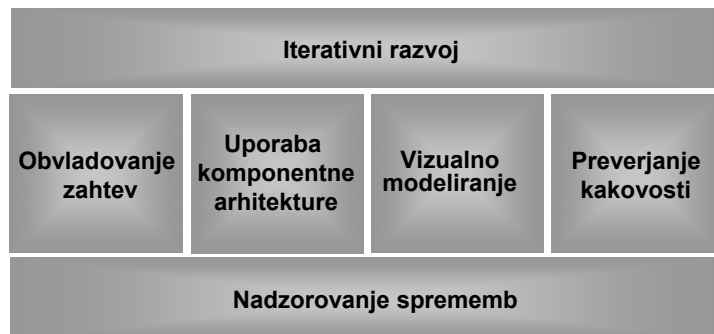
Kaj je proces?

Proces določa **kdo dela kaj, kdaj in kako** za doseganje določenega cilja. Cilj razvoja programske opreme je izgradnja ali izboljšava programskega izdelka.



RUP vpeljuje najboljše izkušnje

RUP opisuje kako učinkovito uporabiti šest najboljših izkušenj s področja razvoja programske opreme



Primeri uporabe so ključen element RUP

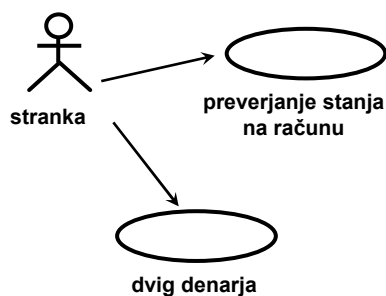




Diagram primera uporabe za bankomat

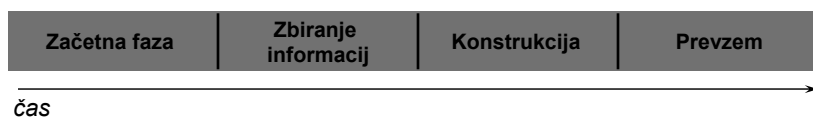
 **Akter** je oseba ali stvar izven sistema, ki je v interakciji s sistemom

 **Primer uporabe** je zaporedje akcij, ki jih izvede sistem in dajo določenemu akterju nek rezultat

Prednosti uporabe primerov uporabe

- Primeri uporabe so jedrnat, enostavni in razumljivi večini udeležencev
 - Končni uporabniki, analitiki in razvijalci razumejo funkcionalne zahteve sistema
- Primeri uporabe so osnova mnogim aktivnostim procesa
 - Izdelava in potrditev razvojnega modela
 - Določitev preizkusnih primerov in postopkov za model preizkušanja
 - Načrtovanje iteracij
 - Izdelava uporabniške dokumentacije
 - vpeljava sistema
- Primeri uporabe pripomorejo k uskladitvi vsebine različnih modelov

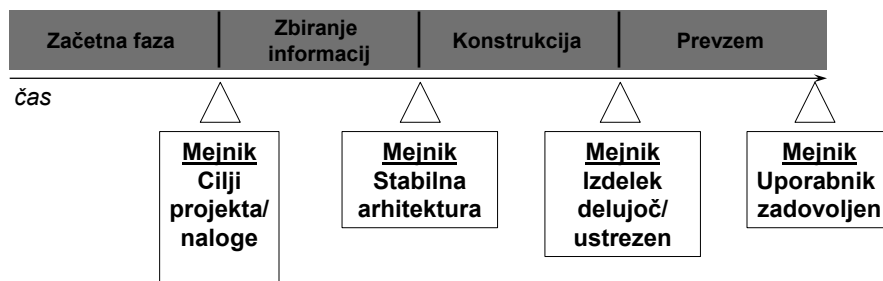
Faze življenjskega cikla razvoja programske opreme



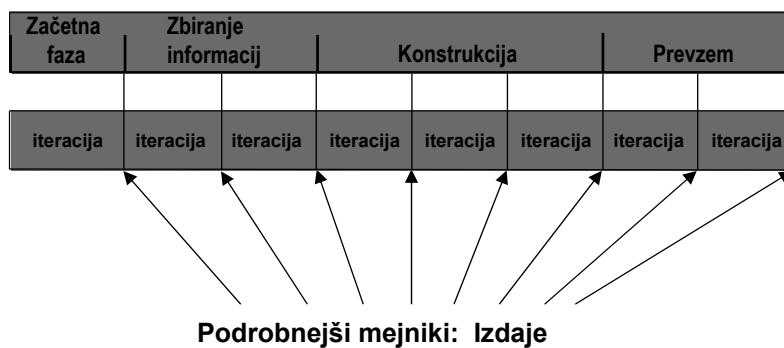
RUP zajema štiri faze:

- **Začetna faza** - vzpostavitev projekta, opredelitev okvirjev obravnavanega področja, načrtovanje virov,...
- **Zbiranje informacij** - zbiranje informacij o obravnavanem področju, specifikacija značilnosti, načrtovanje arhitekture
- **Konstrukcija** - konstrukcija izdelka
- **Prevzem** - predaja izdelka v uporabo končnemu uporabniku

Najpomembnejši mejniki RUP

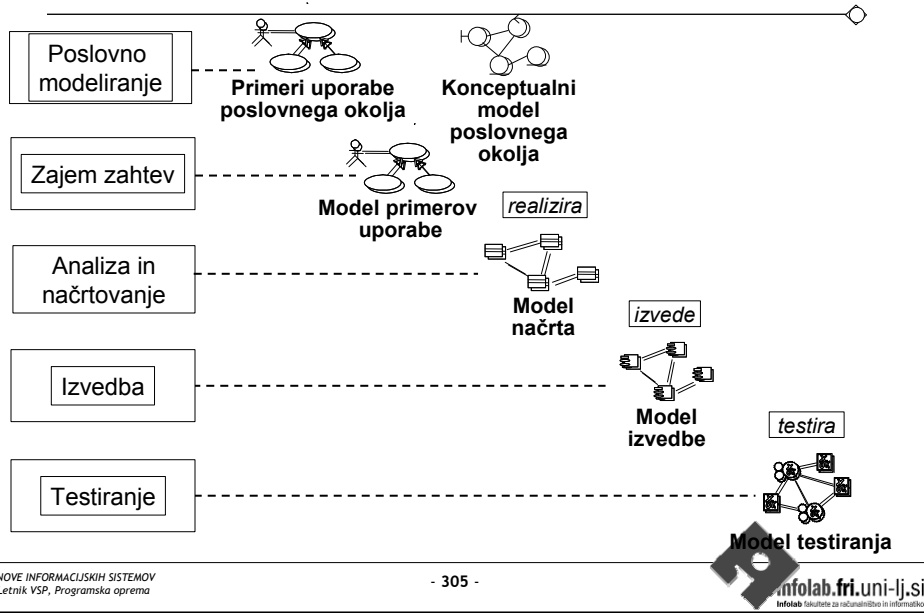


Iteracije in faze življenjskega cikla

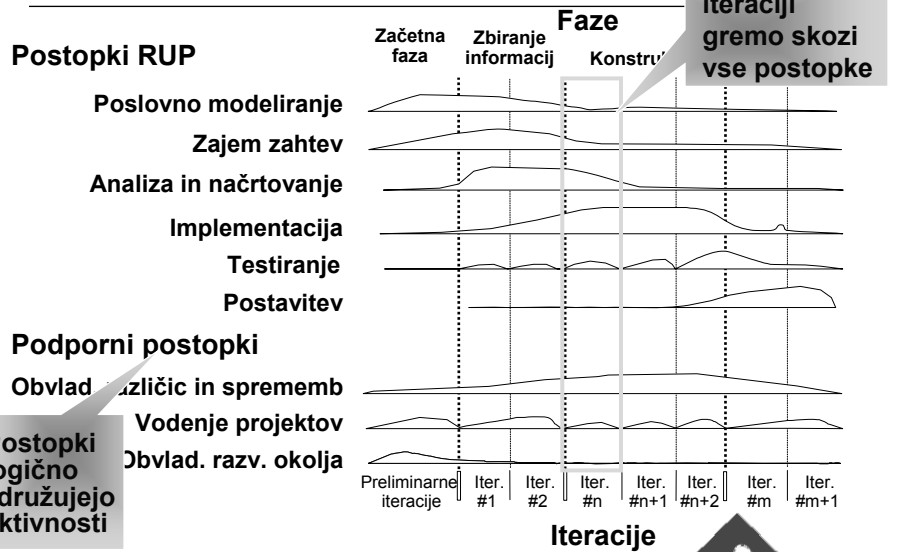


Iteracija je specifično zaporedje aktivnosti izvedenih na osnovi načrta in z določenim kriterijem vrednotenja, ki se konča z izdajo izdelka.

Postopki RUP-a in njihovi modeli 5 - Objektni pristop, RUP



In še vse skupaj: Iterativni model 5 - Objektni pristop, RUP



Notacija procesa



Planiranje virov in dodeljevanje vlog

Vir:	Vloga	Aktivnosti
Miha	Analitik	Analiza operacij sistema
Tine	Snovalec PU	Izdelava primerov uporabe
Jan	Sistemski analitik	Oprelitev akterjev in primerov up.
Sonja	Izvajalec testiranja	Testiranje
Tanja	Arhitekt	Identifikacija mehanizmov za izvedbo

Vsakemu posamezniku, članu projekta, je dodeljena ena ali več vlog

Sklop 5.2 - VSEBINA

- OSNOVE OBJEKTNE USMERJENOSTI
 - Osnovni principi objektne usmerjenosti
 - Osnovni koncepti (gradniki) in izrazi objektne usmerjenosti, povezani z UML notacijo
 - Odlike objektne usmerjenosti
 - Osnovni mehanizmi modeliranja z jezikom UML

Osnovni principi objektne usmerjenosti



Kaj je abstrakcija?



Prodajalec



Stranka

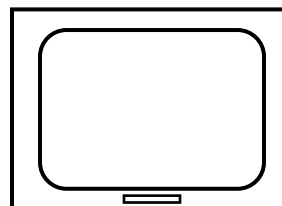
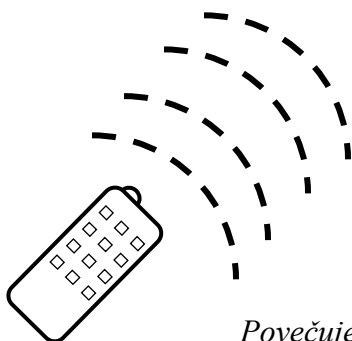


Izdelek

Abstrakcija upravlja s kompleksnostjo

Kaj je ograjevanje?

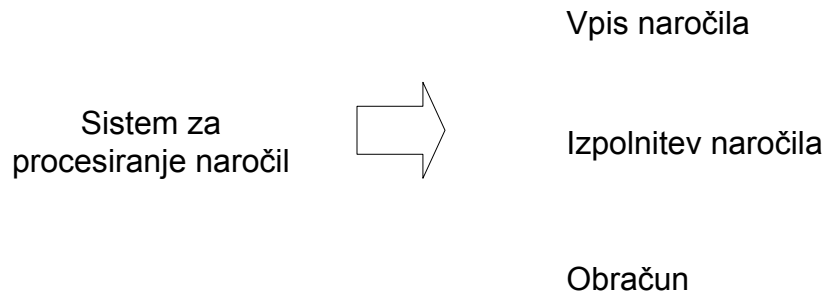
- Skrije posebnosti implementacije pred odjemalcem
 - Odjemalec je odvisen oz. komunicira preko vmesnika



Povečuje prožnost objektov

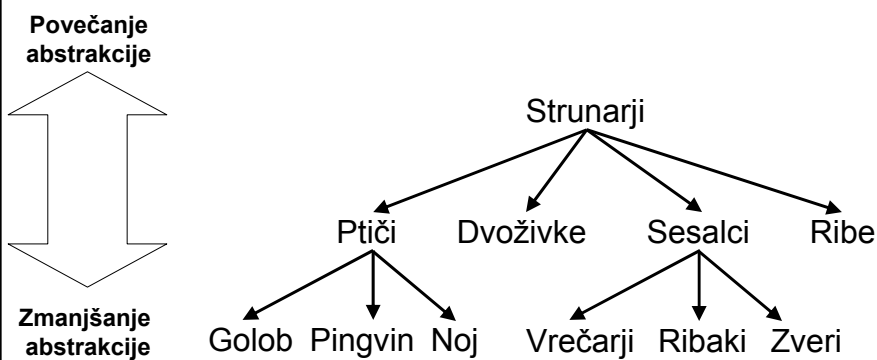
Kaj je modularnost?

- Razbitje nečesa kompleksnega na manjše obvladljive kose (module)



Modularnost upravlja s kompleksnostjo

Kaj je hierarhija?



Elementi na istem nivoju v hierarhiji naj bi imeli enako stopnjo abstrakcije

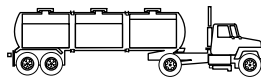
Osnovni koncepti objektne usmerjenosti

- Objekt
- Razred
- Atribut
- Operacija
- Vmesnik (Polimorfizem-mnogoličnost)
- Komponenta
- Paket
- Podsystem
- Povezave

Objekt

- Neformalno, objekt predstavlja fizičen, konceptualni ali programski pojem (entiteto).

- Fizični pojem



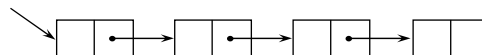
Tovornjak

- Konceptualni pojem



Kemijski proces

- Programski pojem



Povezan seznam

Formalna definicija objekta

- Objekt je koncept, abstrakcija, ali stvar z natančno določenimi mejami in pomenom za aplikacijo
- Objekt je nekaj, kar ima:
 - Stanje
 - Obnašanje
 - Identiteto

Predstavitev objektov

- Objekt je predstavljen s pravokotnikom in podčrtanim imenom

:Profesor

Samo ime razreda

Profesor Korošak:
Profesor

Ime razreda in objekta

Profesor Korošak

Samo ime objekta



Profesor Korošak

(definicija razreda sledi)

Razred

- Razred je opis skupine objektov z enakimi lastnostmi (atributi), enakim obnašanjem (operacije), povezavami, in semantiko (pomenom)
 - Objekt je primerek razreda
- Razred je abstrakcija, ki:
 - Poudarja pomembne karakteristike
 - Izpusti ostale (nepomembne) karakteristike

OO Princip: Abstrakcija

Primer razreda

Razred študij

Lastnosti

ime
lokacija
št. dni
št. ur
začetek
konec

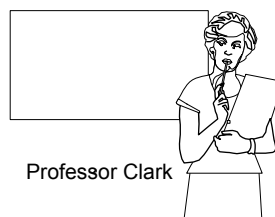
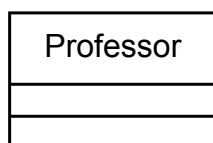


Obnašanje

Dodaj študenta
Zbriši študenta
Preglej seznam študentov
Ugotovi ali je seznam poln

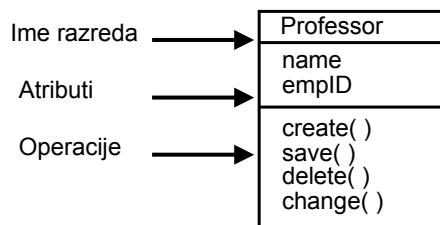
Predstavitev razreda

- Razred je predstavljen s pravokotnikom.



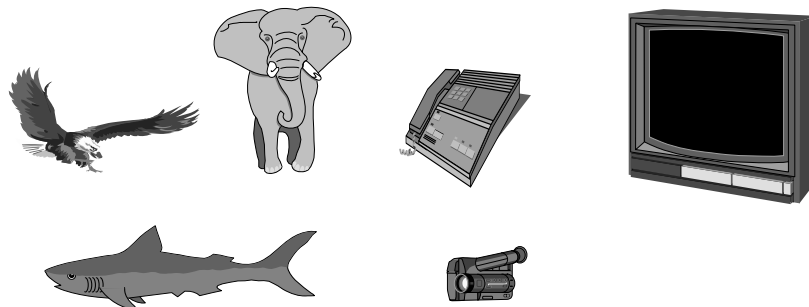
Razdelki v simbolu razreda

- Razred vsebuje tri razdelke
 - Prvi razdelek vsebuje ime razreda
 - Drugi razdelek prikazuje strukturo (atribute)
 - Tretji razdelek prikazuje obnašanje (operacije)



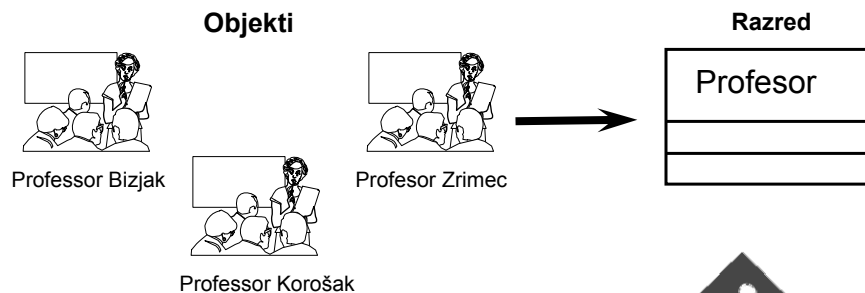
Razredi objektov

- Koliko je razredov?

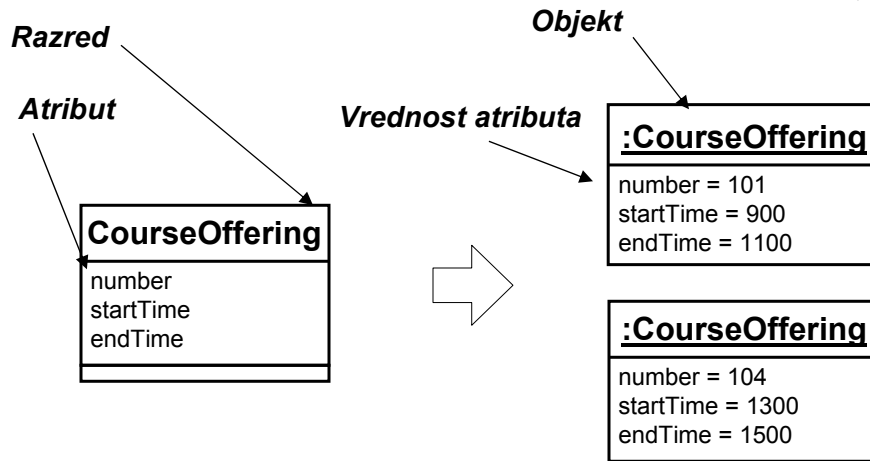


Povezave med razredi in objekti

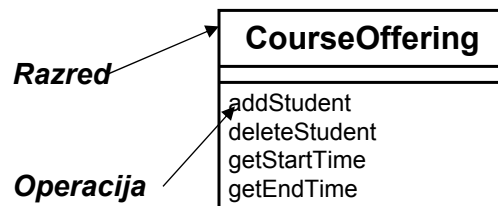
- Razred je abstraktna definicija objekta
 - Definira strukturo in obnašanje vsakega objekta v razredu
 - Služi kot predloga za kreiranje objektov
- Objekti so grupirani v razrede



Atribut

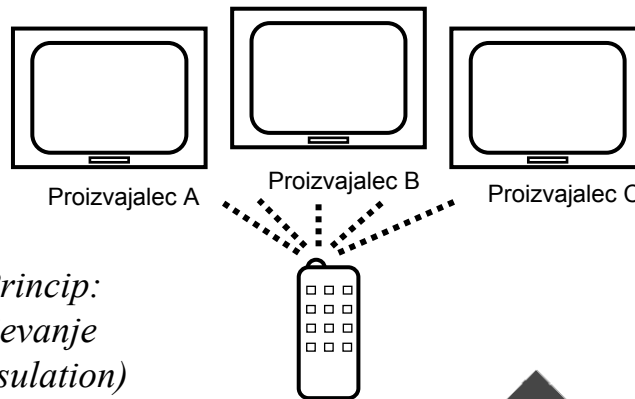


Operacija



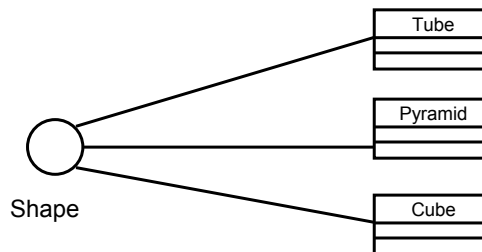
Vmesniki in polimorfizem

- Zmožnost skrivanja več različnih implementacij za enim samim vmesnikom

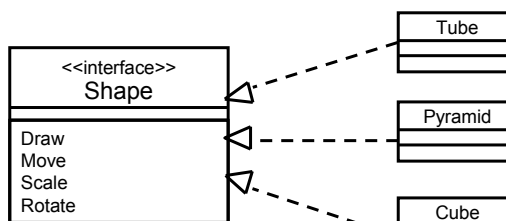


Predstavitev vmesnika

Poenostavljena-
ikonska predstavitev
("lollipop")



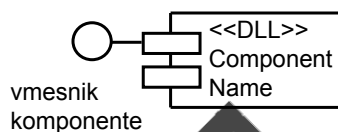
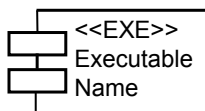
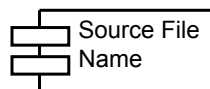
Klasična predstavitev
(razred/stereotip)



Komponenta

- Netrivialen, skoraj neodvisen in nadomestljiv del sistema, ki izpolnjuje jasno funkcijo v skladu z dobro-definirano arhitekturo
- Komponenta je lahko:
 - Komponenta izvorne kode
 - *Run-time* komponenta ali
 - Izvršljiva komponenta

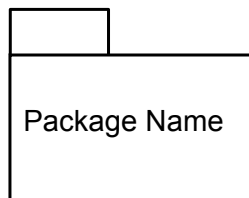
*OO Princip:
Ograjevanje*



vmesnik
komponente

Paket

- Paket je splošen mehanizem za organizacijo elementov modela v skupine
- Je element modela, sestavljen iz drugih elementov modela

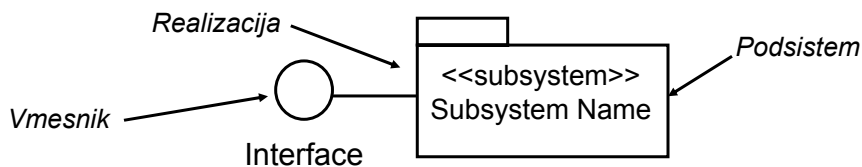


*OO Princip:
Modularnost*

- Uporaba
 - Organizacija modela
 - Paket predstavlja enoto pri upravljanju s konfiguracijo

Podsistem

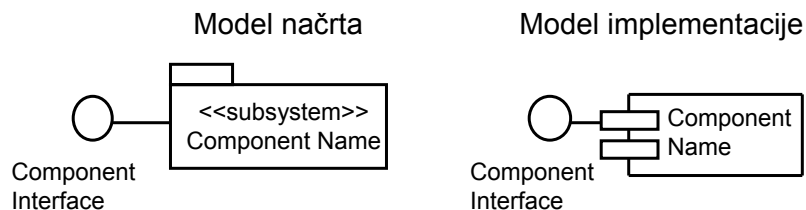
- Podsystem je kombinacija med paketom (vsebuje več drugih elementov modela) in razredom (ima določeno obnašanje)
- Podsystem realizira enega ali več vmesnikov, ki predstavljajo njegovo obnašanje



OO Princip: Ograjevanje in modularnost

Podsistemi in komponente

- Komponente predstavljajo fizično realizacijo abstrakcije iz načrtovanja
- Podsystem lahko uporabimo za predstavitev komponente v času načrtovanja



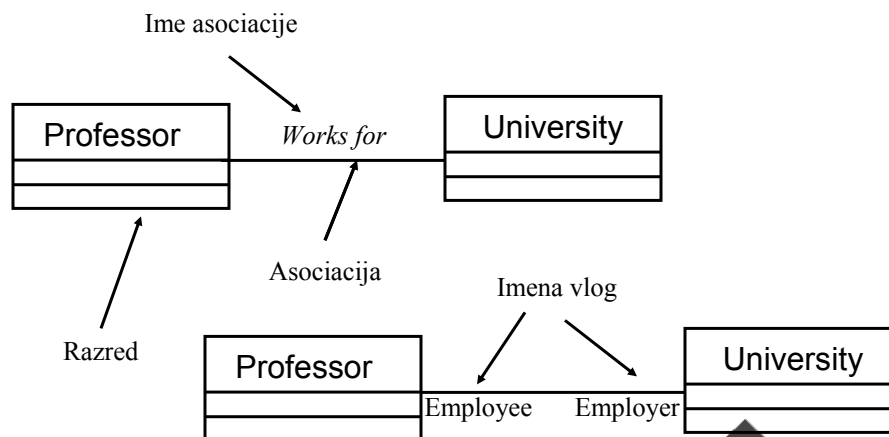
OO Princip: Ograjevanje in modularnost

Povezave

- Asociacija
 - Agregacija
 - Kompozicija
- Odvisnost
- Generalizacija
- Realizacija

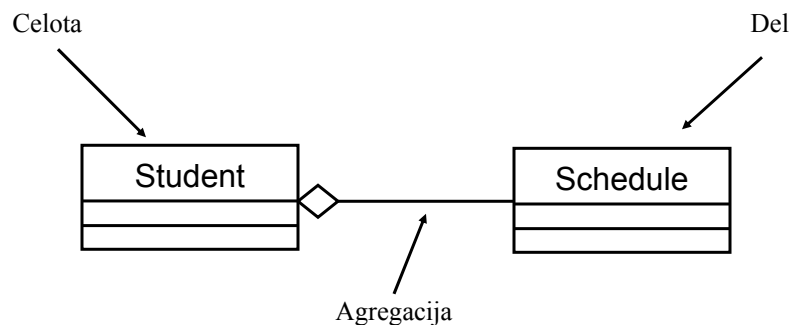
Povezave: Asociacija

- Modelira pomensko povezavo med razredi



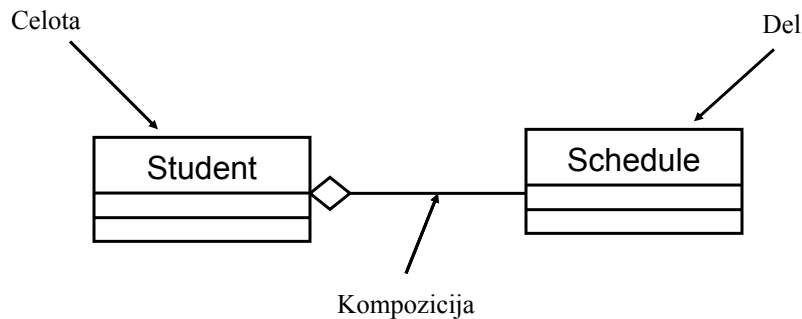
Povezave: Agregacija

- Posebna oblika asociacije, ki modelira povezavo “celota-del” med agregatom (celota) in njenimi deli



Povezave: Kompozicija

- Oblika agregacije z močnim lastništvom in enako življenjsko dobo kot celota
 - Deli ne morejo *živeti* dlje kot celota/agregat



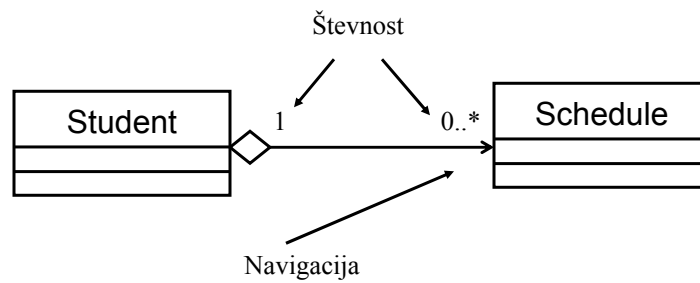
Asociacija: Števnost in navigacija

- Števnost določa koliko objektov je udeleženih v povezavi
 - Število primerkov enega razreda, ki so v povezavi z ENIM primerkom drugega razreda
 - Podana za obe strani (konca) asociacije
- Asociacija in agregacija sta privzeto dvo-smerni, vendar je zaželeno omejiti navigacijo v eno smer
 - Če je navigacija določena, dodamo puščico, ki kaže smer navigacije

Asociacija: Števnost

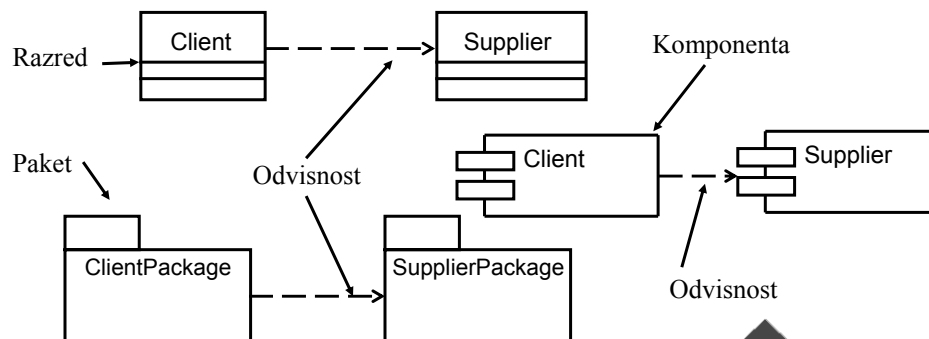
- | | |
|--------------------------------|---------|
| • Nedoločena | _____ |
| • Natanko en | _____ |
| | 1 |
| • Nič ali več (več, neskončno) | _____ |
| | 0..* |
| | _____ |
| | * |
| • Eden ali več | _____ |
| | 1..* |
| • Nič ali eden | _____ |
| | 0..1 |
| • Določeno območje | _____ |
| | 2..4 |
| • Več, razčlenjenih območij | _____ |
| | 2, 4..6 |

Primer: Števnost in navigacija



Povezave: Odvisnost

- Povezava med dvema elementoma modela, kjer lahko sprememba v enem elementu povzroči spremembo v drugem
- Povezava tipa "uporablja"

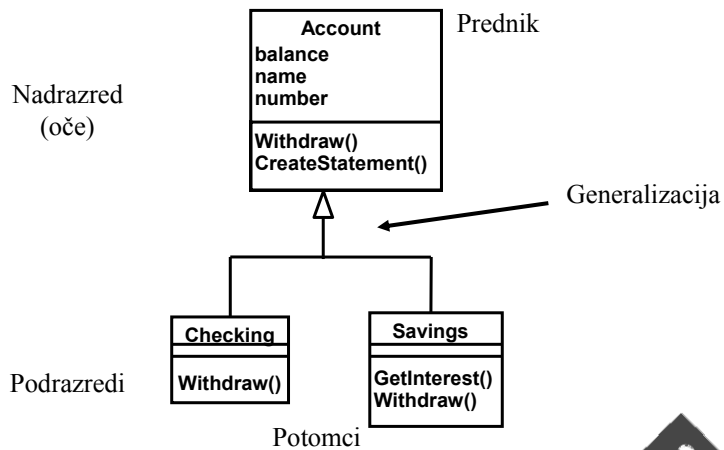


Povezave: Generalizacija

- Povezava med razredi, kjer si en razred deli strukturo in/ali obnašanje enega ali več razredov
- Definira hierarhijo abstrakcije, kjer podrazred deduje od enega ali več nadrazredov
 - Enojno dedovanje
 - Večkratno dedovanje
- Generalizacija je povezava tipa “je vrsta”

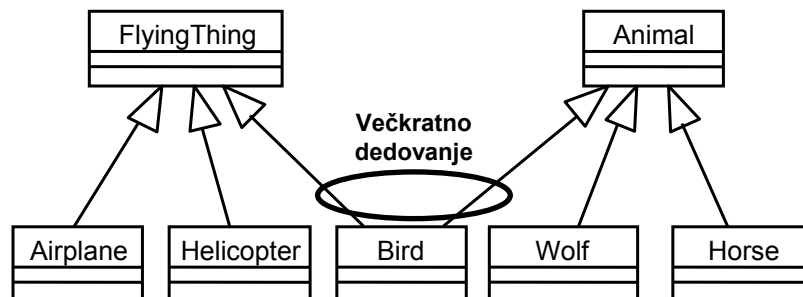
Primer: Enojno dedovanje

- En razred deduje od drugega



Primer: Večkratno dedovanje

- Razred lahko deduje iz več razredov



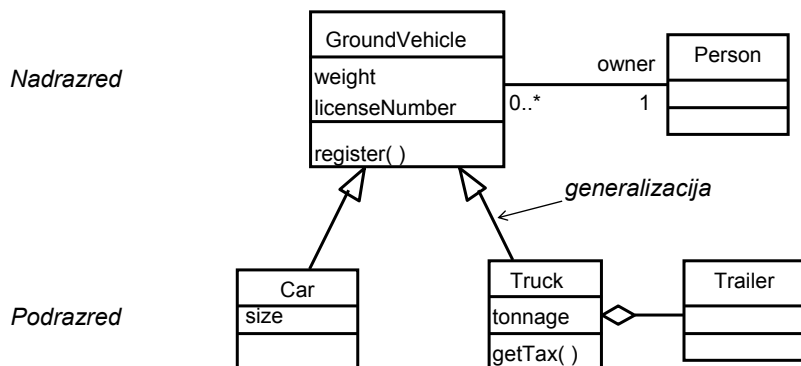
Večkratno dedovanje uporabljamo pazljivo in le ko je to smiselno in potrebno!

Kaj se podeduje?

- Podrazred podeduje attribute, operacije in povezave nadrazreda
- Podrazred lahko:
 - Definira dodatne attribute, operacije in povezave
 - Predefinira podedovane operacije
- Skupni atributi, operacije in/ali povezave so prikazane na najvišjem primernem nivoju v hierarhiji

Dedovanje povzroči nastanek podobnosti med razredi

Primer⁽¹⁾: Kaj se podeduje?



Glavne odlike objektne usmerjenosti

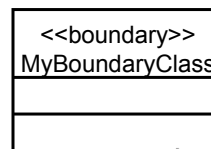
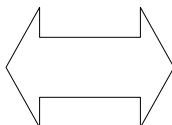
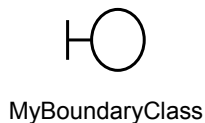
- Ena in enotna paradigma
- Spodbuja ponovno uporabo arhitekture in kode
- Modeli bolj natančno prikazujejo resničen svet
 - Bolj natančno opisujejo podatke in procese v podjetju
 - Razgradnje temeljijo na naravnih delitvah
 - Lažji za razumevanje in vzdrževanje

Osnovni mehanizmi modeliranja z jezikom UML

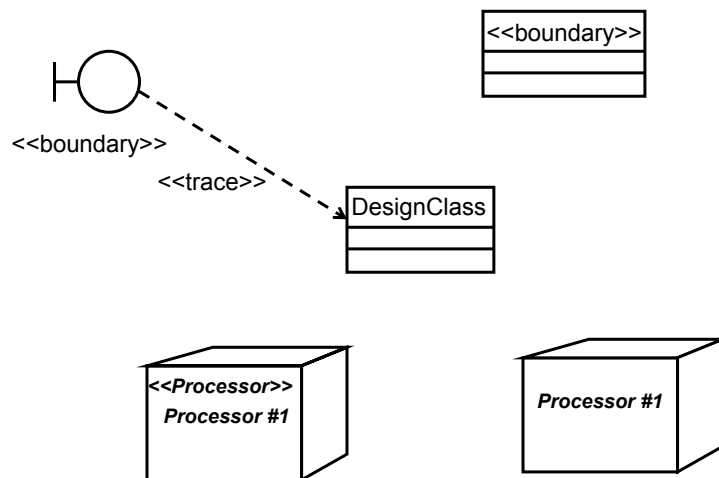
- Stereotip
- Opomba
- Označena vrednost
- Omejitev
- Ogradje
- Vzorec

Stereotip (*stereotype*)

- Razširjajo notacijo jezika UML
- Definirajo nove elemente modela s pomočjo obstoječih elementov
- Lahko jih uporabimo za vse elemente modeliranja
- Predstavimo jih z imenom v dvojnih trikotnih oklepajih ali z novim simbolom

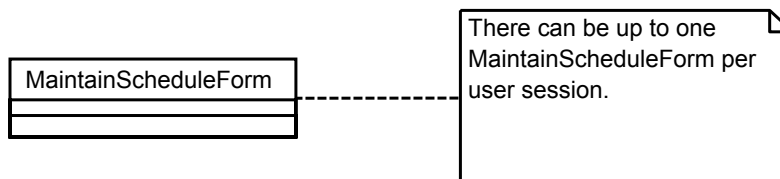


Primer: Stereotip



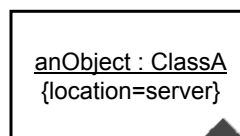
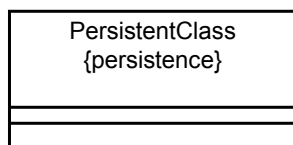
Opomba (note)

- Opombo lahko dodamo kateremkoli UML elementu
- Z dodanimi opombami lahko pripomoremo k informacijski vrednosti diagrama
- Opomba je predstavljena s pravokotnikom z zavihkom
- Opombo povežemo z elementom s prekinjeno črto



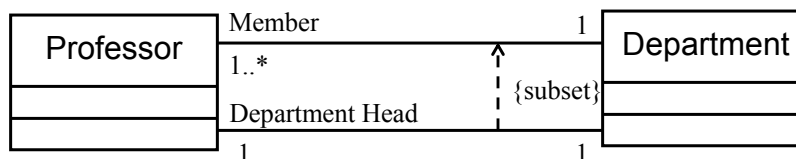
Označena vrednost (*tagged value*)

- Z označenimi vrednostmi razširimo lastnosti ali specifične attribute elementov jezika UML
- Nekaj lastnosti je definiranih že v jeziku UML
 - Trajnost
 - Lokacija (npr.: odjemalec, strežnik)
- Uporabniki UML lahko uporabljajo označene vrednosti v poljubne namene



Omejitev (*constraint*)

- Podpira dodajanje novih pravil ali modifikacijo obstoječih



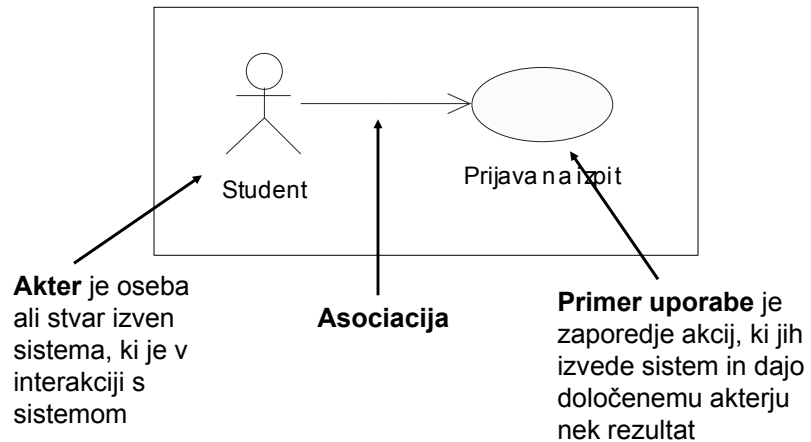
Sklop 5.3 - VSEBINA

- **PODROBNEJE O DIAGRAMIH UML**
 - Diagrami primerov uporabe
 - Razredni diagrami

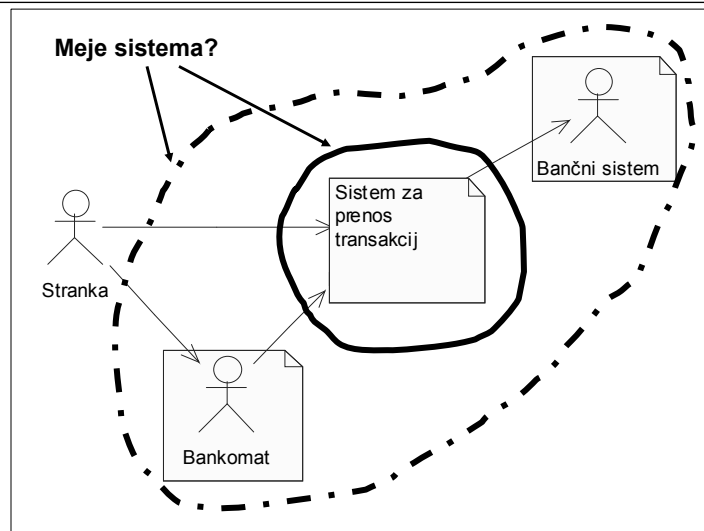
Diagrami primerov uporabe

- Diagram primerov uporabe prikazuje odnos med akterji in primeri uporabe nekega sistema
- Sestavljajo jih trije osnovni elementi
 - akterji
 - primeri uporabe
 - povezave med njimi
- Model primerov uporabe opisuje funkcionalne zahteve sistema v obliki primerov uporabe sistema

Osnovni elementi v diagramu primerov uporabe

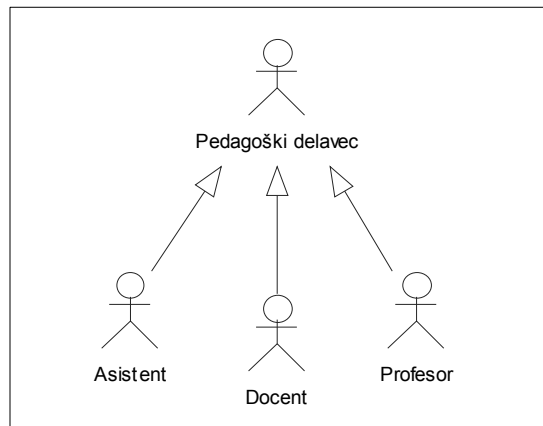


Meje sistema



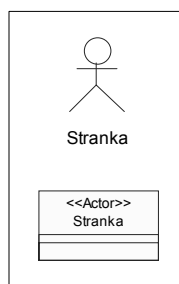
Dedovanje

- Akterje je mogoče specializirati/generalizirati



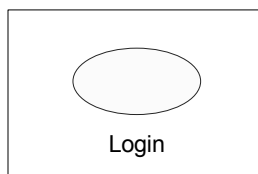
Stereotip akter

- Akter je v orodju Rose predstavljen kot razred z določenim stereotipom



Primer uporabe

- Primer uporabe je zaključen tok dogodkov, ki imajo določen namen
- Primer uporabe prikazuje pomembnejši način uporabe sistema za enega ali več akterjev, ki vplivajo na ta primer uporabe
- Primer uporabe prikazuje določeno funkcionalnost sistema
- V celoti naj bi vsi primeri uporabe vključevali vse možne načine uporabe sistema



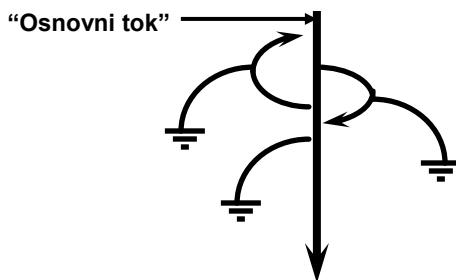
Primer toka dogodkov

Tok dogodkov za primer uporabe - dvig denarja

1. Primer uporabe se prične z vstavitvijo bančne kartice. Sistem prebere in potrdi informacije na kartici.
2. Sistem zahteva vnos PIN kode. Sistem potrdi PIN.
3. Sistem zahteva izbiro operacijo, ki jo stranka želi izvesti. Stranka izbere "dvig denarja".
4. Sistem zahteva vnos vrednosti. Stranka vnese vrednost.
5. Sistem komunicira s sistemom za prenos transakcij ...

Primer uporabe ima lahko več tokov dogodkov

- Primer uporabe ima en normalen oz. osnovni tok ter več alternativnih tokov:
 - Običajni primeri
 - Nenavadni primeri
 - Izjemni primeri, napake



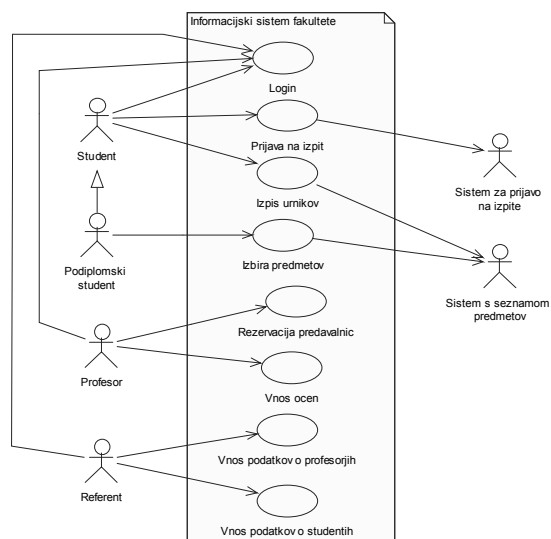
Asociacije

- Asociacije so lahko različnih stereotipov
- Osnovni stereotipi asociacij pri primerih uporabe so
 - komunicira <<communicate>>
 - razširja <<extend>>
 - vsebuje <<include>>

Asociacija <<komunicira>>

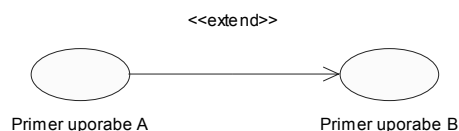
- Ta asociacija je v orodju Rose privzeta in povezuje akterja s primerom uporabe
- Usmerjenost povezave:
 - kadar gre puščica iz akterja proti primeru uporabe to pomeni, da je akter začetnik primera uporabe (uporablja določeno funkcionalnost sistema)
 - kadar gre puščica iz primera uporabe proti akterju (ali pa puščic ni) to pomeni, da ti akterji sodelujejo pri določenem primeru uporabe
 - vsak primer uporabe mora imeti povezavo, ki je usmerjena od akterja proti primeru uporabe

Asociacija <<komunicira>> - primer

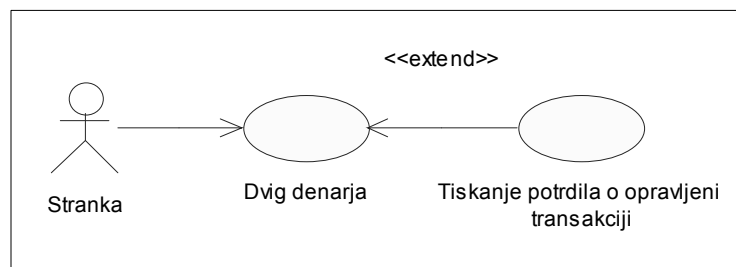


Asociacija <<razširja>>

- Asociacija <<razširja>> med seboj povezuje primere uporabe
- Povezava je usmerjena od primera uporabe, ki zagotavlja razširitev proti osnovnemu primeru uporabe
- Na sliki vidimo *Primer uporabe A*, ki razširja tok dogodkov *Primer uporabe B*

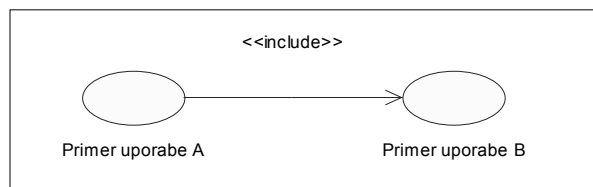


Asociacija <<razširja>> - primer

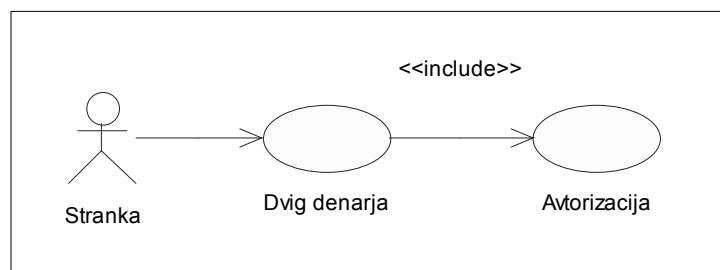


Asociacija <<vsebuje>>

- Asociacija <<vsebuje>> med seboj povezuje primere uporabe
- Povezava je usmerjena od primera uporabe, ki vsebuje tudi drug primer uporabe
- Na sliki vidimo primer uporabe A, ki v svojem delovanju vsebuje tudi delovanje primera uporabe B



Asociacija <<vsebuje>> - primer



Sklop 5.4

- Diagrami primerov uporabe
- ⇒ • Razredni diagrami

Razredni diagrami

- Razredni diagrami prikazujejo statično strukturo modela sistema
- Prikazujejo razrede, njihovo strukturo, metode, attribute
- Ne prikazujejo dinamičnih informacij oziroma stvari, ki opisujejo časovno obnašanje

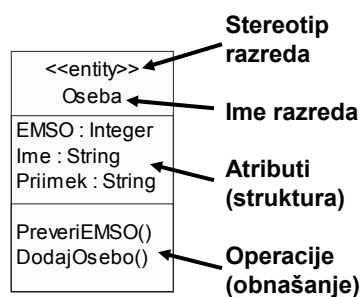
Kaj določimo razredu?

- Ime
- Stereotip
- Atribute in operacije
- Dodatne lastnosti
 - Vidljivost
 - Trajnost
 - Sočasnost
- Tip razreda
- Povezave

Razred

- Razred je opis skupine objektov z enakimi lastnostmi (atributi), enakim obnašanjem (operacije), povezavami, in semantiko (pomenom)
 - Objekt je primerek razreda
- Razred je abstrakcija, ki:
 - Poudarja pomembne karakteristike
 - Izpusti ostale (nepomembne) karakteristike

Prikaz razreda v orodju Rose



Stereotip razreda

- Stereotip razširja osnovni pomen elementa modela, oziroma definira nove elemente na podlagi že obstoječih.
- Osnovni stereotipi razredov so:
 - Mejni razred (<<*Boundary*>>)
 - Kontrolni razred (<<*Control*>>)
 - Poslovni razred (<<*Entity*>>)
- Stereotipi razredov so tudi elementi kot so akterji in vmesniki

Kaj določimo atributom?

- Ime
- Tip
- Stereotip
- Začetno vrednost
- Vidljivost
- Vsebovanost in druge lastnosti

Kaj določimo operacijam?

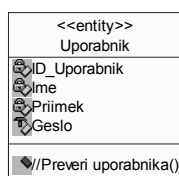
- Ime
- Tip vrednosti, ki jo vrne funkcija
- Stereotip
- Vidljivost
- Predpogoji
- Popogoji
- Sočasnost in druge lastnosti

Vidljivost atributov in operacij

- Vidljivost atributov in operacij določa kateri razredi lahko dostopajo do atributov in operacij določenega razreda.
- Tipi vidljivosti:
 - Javna
 - Zaščiten
 - Zasebna
 - Izvedbena

Oznake vidljivosti atributov in operacij

Tip vidljivosti	Notacija UML	Notacija Rose (atributi)	Notacija Rose (operacije)
Javna	+		
Zaščitena	#		
Zasebna	-		
Izvedbena	Ni del UML		



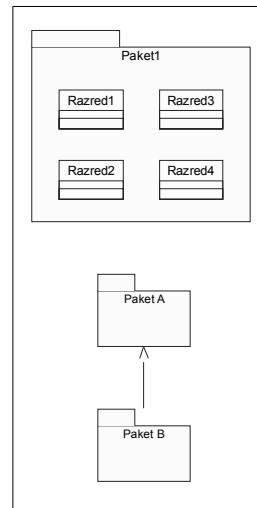
Prikaz oznak vidljivosti atributov in operacij v orodju Rose

Dodatne lastnosti atributov

- Atributom lahko določimo, da so:
 - Statični (*Static*)
 - Izpeljani (*Derived*)
- Vsebovanost atributov:
 - Po sklicu (*By reference*)
 - Po vrednosti (*By value*)

Razrede združujemo v pakete

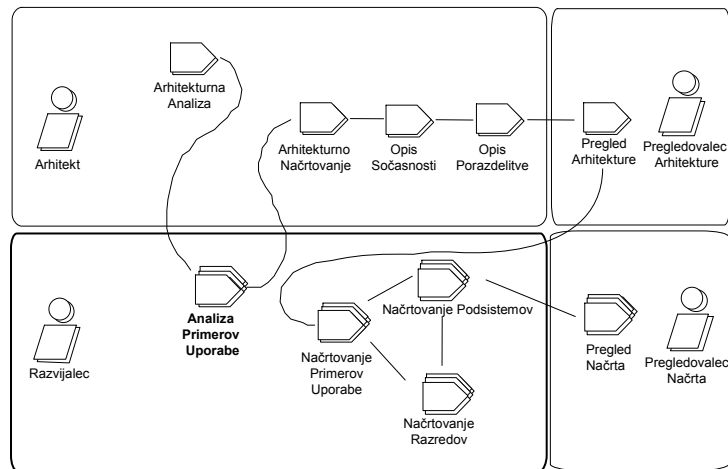
- Paket združuje različne elemente modela - tudi razrede
- Paketi so lahko odvisni drug od drugega (na sliki je Paket B odvisen od Paketa A)



Postopek analize in načrtovanja

- 2. aktivnost: ANALIZA PRIMEROV UPORABE

Analiza primerov uporabe

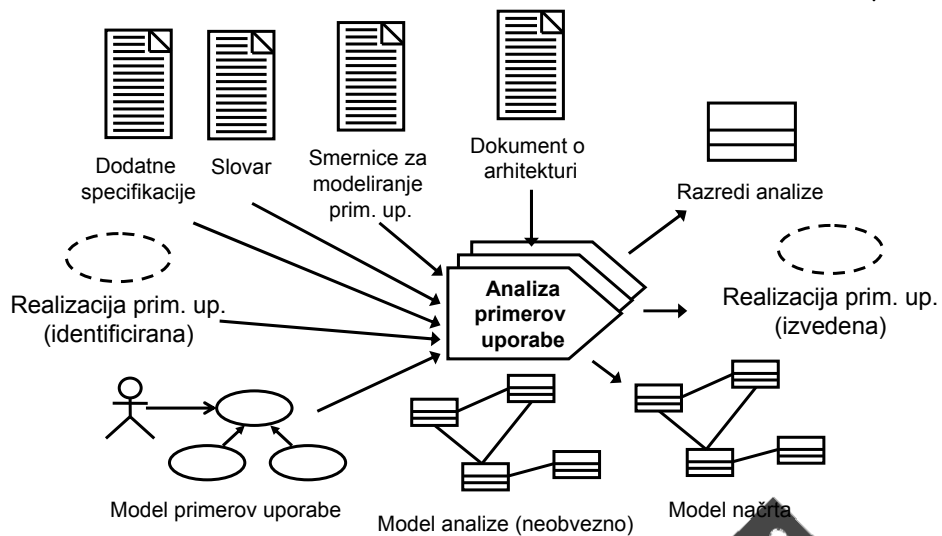


OSNOVE INFORMACIJSKIH SISTEMOV
2. Letnik VSP, Programska oprema

- 381 -



Analiza primerov uporabe - pregled



OSNOVE INFORMACIJSKIH SISTEMOV
2. Letnik VSP, Programska oprema

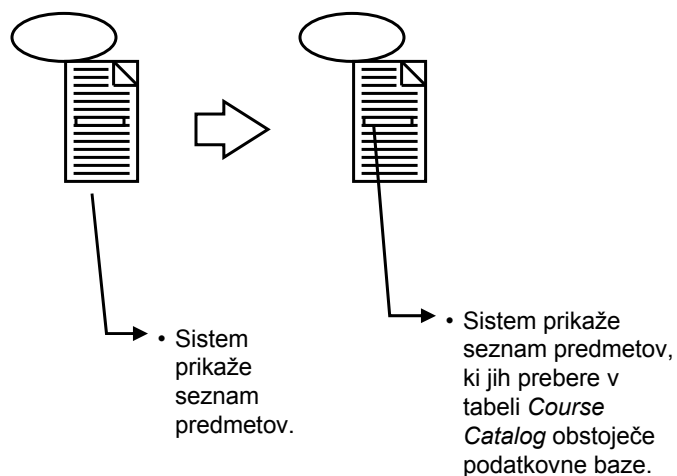
- 382 -



Koraki analize primerov uporabe

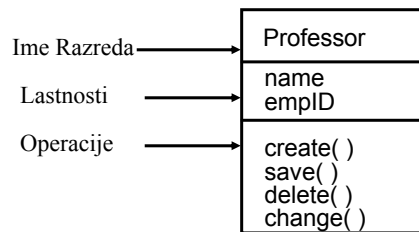
- Dopolni opise primerov uporabe
- Za vsako realizacijo primera uporabe
 - Na osnovi dinamike primera uporabe identificiraj potrebne razrede
 - Posameznim razredom dodeli odgovornosti v zvezi z dinamiko primera uporabe
- Za vsak razred pridobljen z analizo
 - Opiši odgovornosti
 - Opiši attribute in povezave
 - Podrobneje analiziraj potrebne sistemske storitve
- Poenoti razrede pridobljene z analizo

Dopolnitev opisov primerov uporabe



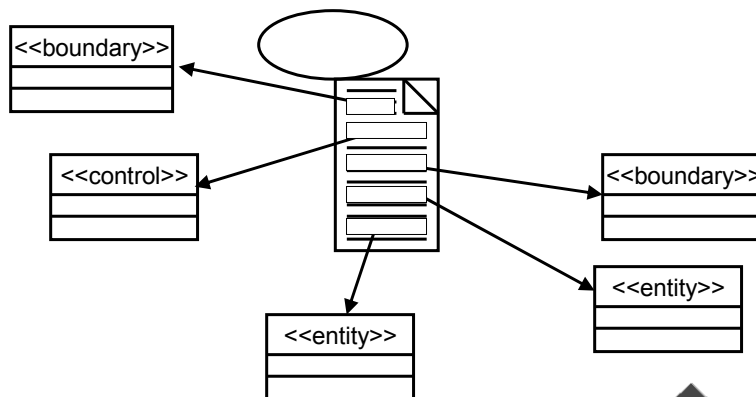
Ponovitev: Razred

- Abstrakcija
- Združuje objekte, ki imajo:
 - Skupne lastnosti (atribute)
 - Skupno obnašanja (operacije)
 - Skupne povezave
 - Skupen Pomen

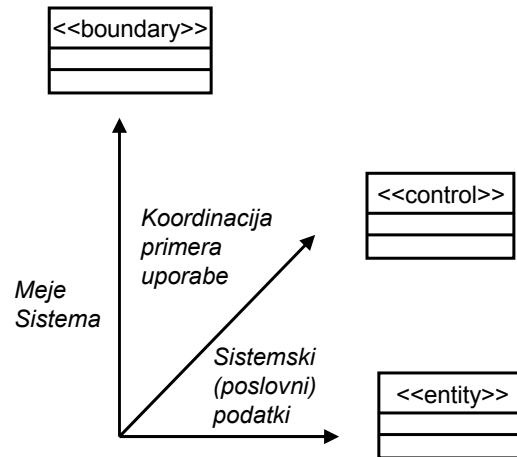


Identifikacija razredov

- Dinamika primera uporabe mora biti v celoti dodeljena (pokrita) z razredi analize

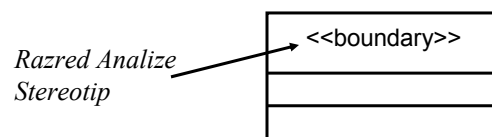


Kaj je Razred Analize?



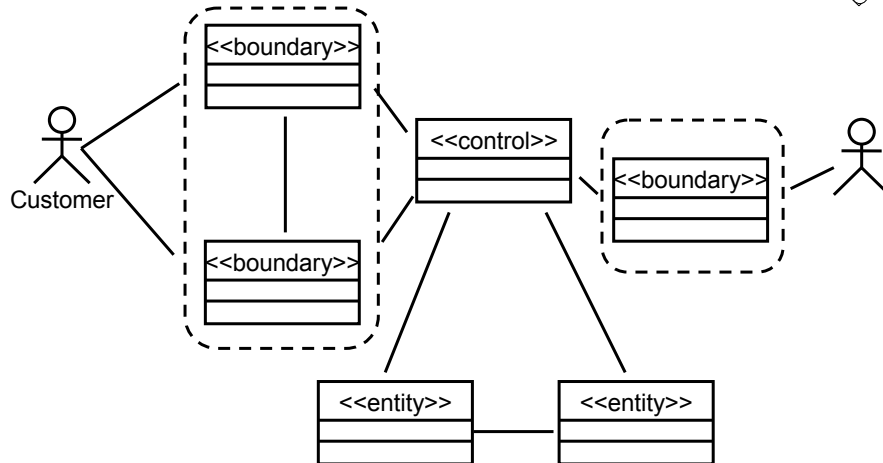
Kaj je Mejni Razred (*Boundary Class*)?

- Posrednik med okoljem in sistemom
- Več tipov
 - Razredi uporabniškega vmesnika
 - Razredi sistemskega vmesnika
 - Razredi vmesnika do naprav
- En mejni razred za vsak par akter/primer uporabe



Odvisen od okolja

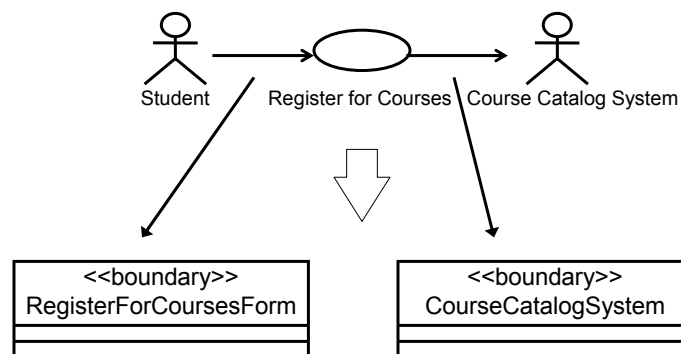
Kakšna je vloga Mejnega Razreda?



Modelira interakcijo med okoljem in sistemom

Primer: Identifikacija mejnih razredov

- En mejni razred za vsak par akter/primer uporabe



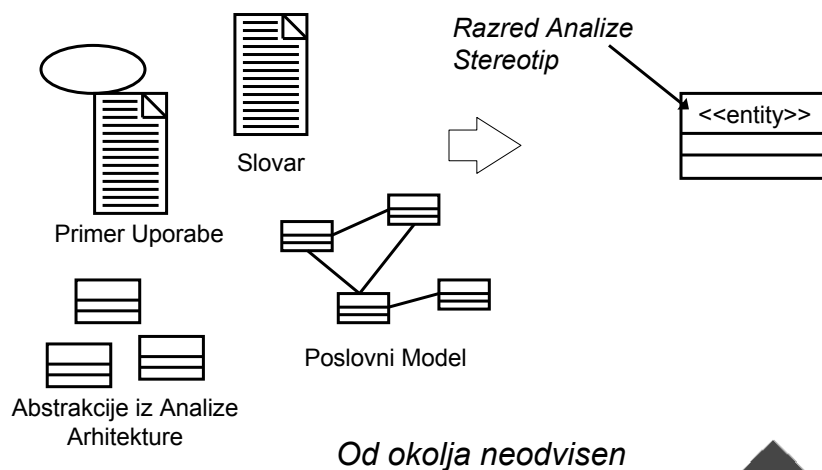
Smernice: Mejni razredi

- Razredi uporabniškega vmesnika
 - Osredotoči se na vprašanje: kakšni podatki so posredovani uporabniku? Kakšne storitve mora sistem nuditi uporabniku?
 - Ne ukvarjaj se s podrobnostmi uporabniškega vmesnika
- Razredi systemskega vmesnika in vmesnika do naprav
 - Osredotoči se na potrebne protokole za komunikacijo. Kakšne storitve mora sistem nuditi drugim sistemom?
 - Ne ukvarjaj se z vprašanjem, kako bodo protokoli dejansko izvedeni

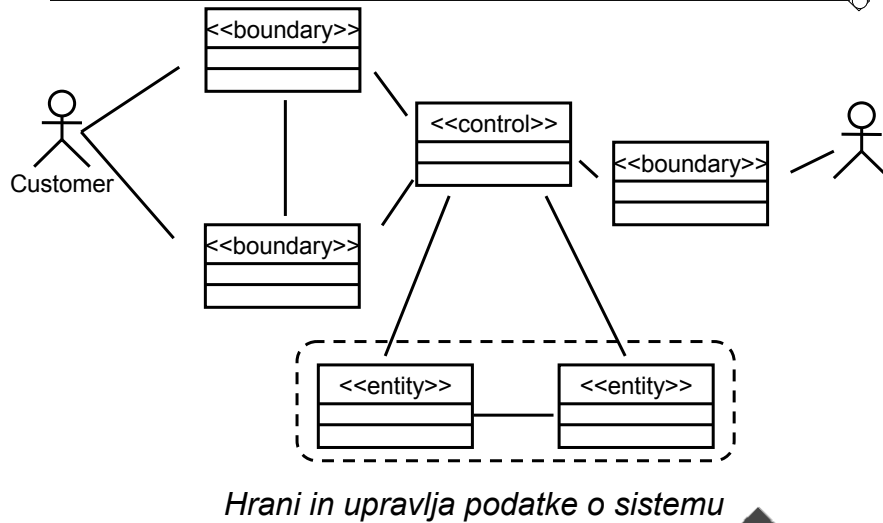
Osredotoči se na odgovornosti, spusti podrobnosti!

Kaj je Poslovni razred (*Entity Class*)?

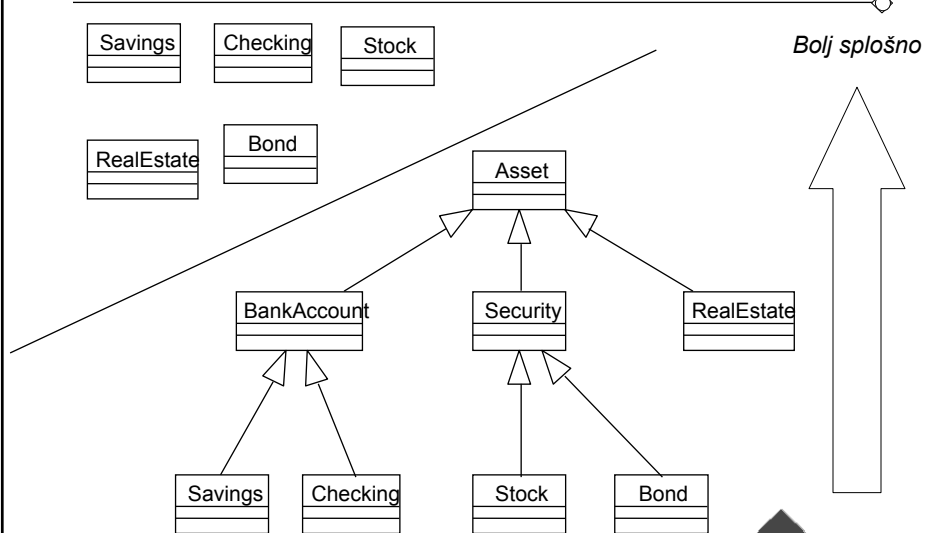
- Ključna abstrakcija sistema



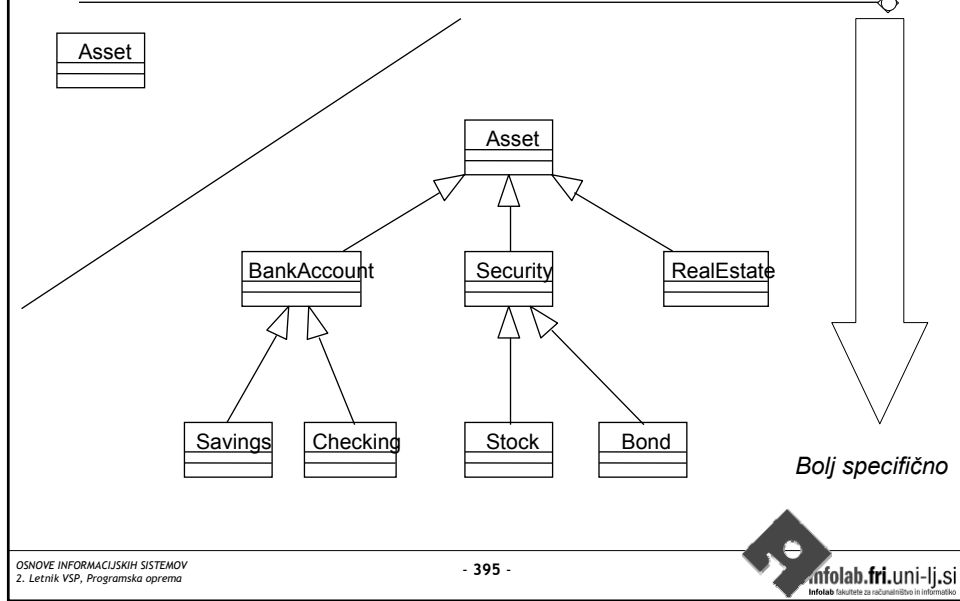
Kakšna je vloga Poslovnega razreda?



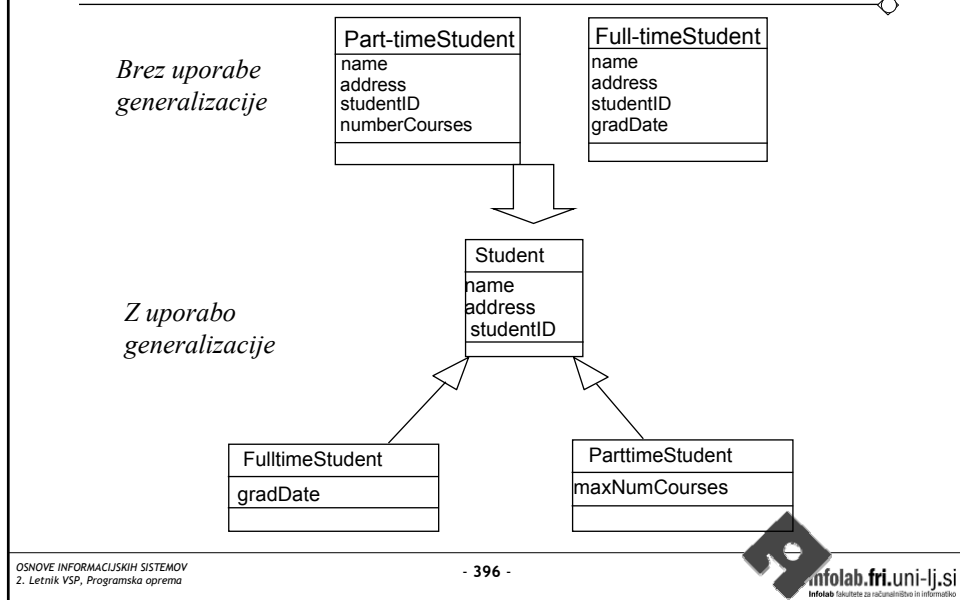
Identifikacija generalizacije razredov



Identifikacija generalizacije razredov

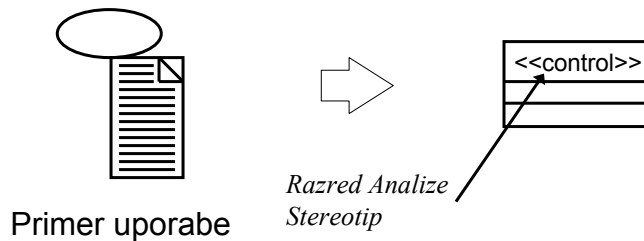


Primer: Generalizacija (Skupen pomen)



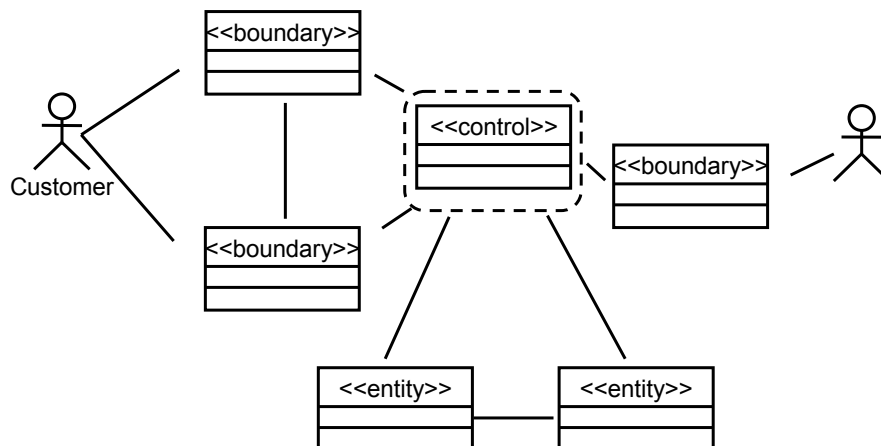
Kaj je Kontrolni razred (*Control Class*)?

- Koordinator dinamike primera uporabe
- En kontrolni razred za vsak primer uporabe



Odvisen od primera uporabe, neodvisen od okolja

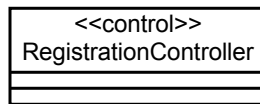
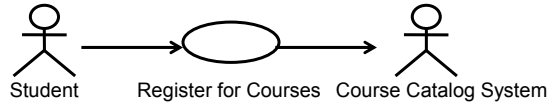
Kakšna je vloga Kontrolnega razreda?



Koordinira dinamiko oziroma obnašanje primera uporabe

Identifikacija kontrolnih razredov

- En kontrolni razred za vsak primer uporabe



Primer razrednega diagrama

5 - Objektni pristop, RUP

