

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

DIPLOMSKO DELO
PODATKOVNI MODEL PROIZVODNEGA PODJETJA

Ljubljana, september 2005

JOŽKO HOČEVAR

IZJAVA

Študent JOŽKO HOČEVAR izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom doc. dr. MOJCE INDIHAR ŠTEMBERGER in dovolim objavo diplomskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne _____

Podpis: _____

KAZALO

1. UVOD.....	1
2. RAZVOJ INFORMACIJSKIH SISTEMOV.....	2
2.1. Pristopi k razvoju informacijskih sistemov.....	2
2.1.1. Življenjski cikel sistema.....	3
2.1.2. Alternativni pristopi gradnje informacijskih sistemov.....	3
2.1.2.1. Razvoj s prototipom.....	3
2.1.2.2. Razvoj s strani končnih uporabnikov.....	4
2.1.2.3. Uporaba programskih paketov..	5
2.1.2.4. Zunanje izvajanje (Outsourcing).....	6
3. MANAGEMENT PODATKOV.....	6
3.1. Komponente okolja podatkovnih baz.....	7
3.1.1. Uporabniki.....	7
3.1.2. Uporabniški vmesnik.....	8
3.1.3. Podatkovna baza.....	8
3.1.4. Podatkovni slovar.....	10
3.1.5. Skrbništvo podatkovne baze.....	11
3.1.6. Sistem za upravljanje podatkovnih baz.....	12
4. PODATKOVNO MODELIRANJE.....	14
4.1. Analiza podatkovnih potreb.....	15
4.2. Modeli podatkov.....	16
4.2.1. Trinivojska arhitektura.....	16
4.2.2. Modeli podatkov odvisni od uporabljene tehnologije.....	17
4.2.2.1. Starejši podatkovni modeli.....	18
4.2.2.2. Relacijski model podatkov.....	19
4.2.2.3. Večdimenzionalni model podatkov.....	21
4.2.2.4. Objektni model podatkov.....	22
4.2.2.5. Objektno-relacijski model podatkov.....	25
4.2.3. Tehnološko neodvisni modeli podatkov.....	26
4.2.3.1. Model entitet-povezav.....	26
4.2.3.2. Semantični model.....	31
5. PODATKOVNI MODEL PROIZVODNEGA PODJETJA.....	32
5.1. Kratak opis podjetja.....	32
5.2. Globalni podatkovni model podjetja.....	33
5.3. Relacijska shema.....	38
5.4. Prototip predlagane podatkovne baze.....	40
5.5. Posledice upoštevanja izsledkov obravnavane tematike.....	43
6. SKLEP.....	43
LITERATURA.....	45
VIRI.....	45

1. UVOD

Danes, bolj kot kdajkoli, prihajamo do spoznanja, da je potreben celovit pristop k izgradnji informacijskih sistemov, ki bo organizacijam omogočil hitro prilagajanje spreminjajočim se pogojem delovanja. Te spremembe so posledica dinamike okolja, tehnološkega napredka na področju računalniške in komunikacijske tehnologije in spreminjanja vlog, ki jih imajo posamične skupine uporabnikov pri načrtovanju, implementaciji in sami uporabi informacijskega sistema. Spoznanje, da so podatki eden ključnih virov organizacije, s pomočjo katerih lahko ta dosega konkurenčno prednost, nas pripelje do poskusa postaviti globalni podatkovni model, ki bo v svojem bistvu zajemal in zadovoljeval potrebe po informacijah organizacije danes in jutri.

Namen mojega diplomskega dela je preučiti vse dosedanje ugotovitve s področja managementa podatkov in dejavnikov, ki vplivajo na dinamiko spreminjajočih se pogojev delovanja, kakšen ima to vpliv na zahteve informacijskega sistema in nenazadnje, tako poglobljeno znanje uporabiti pri postavitvi globalnega podatkovnega modela na primeru dejanskega proizvodnega podjetja.

Tako se v prvi točki dotaknem informacijskih sistemov z vidika njihovega pomena za vsako organizacijo. Izpostavljeni so različni pristopi k gradnji informacijskih sistemov, ki so v veliki meri posledica tako evolucije spoznanj o pomenu podatkov kot ključnega dejavnika in vira poslovanja kot tudi razvoja tehnologije, ki je omogočila implementacijo teh spoznanj. V drugi točki je govor o managementu podatkov, torej o podatkovni bazi, kot osrednjem elementu vsakega informacijskega sistema, ki je nastala kot posledica dojemanja edinstvenosti podatkov med viri organizacije, saj je od njihove dostopnosti odvisna učinkovitost upravljanja z vsemi ostalimi viri organizacije. V podrobnostih so obrazložene vse komponente okolja podatkovnih baz. V tretji točki je obravnavana problematika modeliranja podatkov. Od začetnih faz, to je analiza podatkovnih potreb, do samega modeliranja ugotovitev, pri čemer se najprej sprehodimo čez starejše podatkovne modele, ki so odvisni od uporabljene tehnologije, do malo podrobnejših razlag danes uporabljenih modelov, kot so relacijski, večdimenzionalen, objektni in tako naprej. V podrobnostih sta opisana tudi tehnološko neodvisna podatkovna modela, in sicer model entitet-povezav in semantični model.

V četrti točki je prikazan moj poskus implementacije dognanj tega diplomskega dela na praktičnem primeru. Podan je kratek opis podjetja, postavljen pa je tudi globalni podatkovni model, ki upošteva potrebe po informacijah, ugotovljenih s pomočjo intervjujev zaposlenih in analize obstoječe dokumentacije obravnavanega podjetja. Z izdelavo relacijske sheme so podana tudi sama izhodišča prototipa podatkovne baze, ki bo upoštevala specifiko podjetja na eni strani in specifiko uporabljene tehnologije na drugi.

2. RAZVOJ INFORMACIJSKIH SISTEMOV

Zaradi hitrega povečevanja zmogljivosti ter vedno večje dostopnosti informacijske in komunikacijske tehnologije, nenadnih in nepričakovanih sprememb v okolju organizacije, je potrebno tudi upoštevanje stalnega prilagajanja informacijskih sistemov. Ti predstavljajo pomemben element konkurenčne prednosti, zato mora biti njihovo načrtovanje tesno povezano in usklajeno s cilji organizacije.

Pri razvijanju informacijskih sistemov sodelujejo poklicni informatiki, bodoči uporabniki in managerji. Managerji določijo okvirno vsebino bodočega sistema, ki mora biti usklajena s strateškimi in taktičnimi načrti organizacije. Poleg tega spremljajo razvoj in pomagajo pri reševanju nepredvidenih težav. Bodoči uporabniki pri razvoju sodelujejo tako, da do najmanjših podrobnosti določijo svoje zahteve in s tem natančno opredelijo vsebino informacijskega sistema. Naloga informatikov pa je, da izberejo ustrezno računalniško in komunikacijsko tehnologijo, zasnujejo bazo podatkov in izdelajo računalniške programe tako, da bo delovanje informacijskega sistema čim bolj hitro, zanesljivo in poceni (Gradišar, Resinovič, 1996, str. 413). Vendar pa vloga informatikov ni omejena le na izvajanje, ampak sodelujejo tudi pri procesu strateškega načrtovanja informatike in prenovi procesov.

2.1. PRISTOPI K RAZVOJU INFORMACIJSKIH SISTEMOV

Začetek večine informacijskih sistemov je nastanek nekega poslovnega problema ali priložnosti, ki povzroči idejo o novem sistemu, s katerim bi ta problem rešili oziroma izkoristili priložnost. Naslednji korak je razvijanje sistema, ko je le-ta izdelan, ga je potrebno uvesti v prakso. Nekaj časa obratuje poskusno, nato pa redno. V toku rednega obratovanja oziroma izvajanja je potrebno sistem vzdrževati, kar pomeni prilagajanje spremembam, ki se dogajajo v organizaciji in njenem okolju. Sčasoma postanejo te spremembe tako velike, da nadaljnje prilagajanje sistema ni več smotno, postane zastarel. Zamenjati ga je potrebno z novim. Opisan splošen model načrtovanja in razvoja informacijskih sistemov predstavlja življenjski krog sistema, ki ga sestavljajo štiri osnovne faze, ki jih vsebuje katerikoli pristop razvoja informacijskih sistemov, najbolj izrazite pa so pri modelu življenjskega cikla (Gradišar, Resinovič, 1996, str. 413-414):

- začetek,
- razvoj,
- uvajanje,
- vzdrževanje.

2.1.1. ŽIVLJENJSKI CIKEL SISTEMA

Gre za najstarejšo metodo gradnje informacijskih sistemov, ki pa je še vedno v uporabi tudi danes za srednje in velike kompleksne projekte. Metodologija predpostavlja, da ima informacijski sistem življenjski cikel podoben kateremukoli živemu organizmu. Življenjski cikel informacijskega sistema ima šest stopenj (Laudon, Laudon, 2000, str. 370-372):

1. Definiranje projekta (angl. project definition): v tej fazi se izkaže, ali ima organizacija problem, ki bi se ga dalo razrešiti z izgradnjo novega informacijskega sistema oziroma s spreminjanjem obstoječega. Če je temu tako, identificira splošne cilje, določi obseg projekta in razvije načrt projekta, ki se ga lahko predstavi managementu.
2. Analiza sistema (angl. system study): je faza, v kateri analiziramo probleme obstoječega sistema, definiramo cilje, ki naj bi jih dosegli z rešitvijo in ocenjujemo številne alternativne rešitve. Zahteva zbiranje podrobnih informacij in raziskovanje s pomočjo pregledovanja dokumentov, poročil in druge dokumentacije v obstoječem sistemu, anketiranjem in izvajanjem intervjujev uporabnikov, itd.
3. Načrtovanje (angl. design): ta faza rezultira v logičnih in fizičnih oblikovnih specifikacijah predlaganega sistema. Uporabljajo se orodja za oblikovanje in dokumentiranje, kot so diagram toka podatkov (angl. data flow diagram), model entitet-povezav (ER diagram), procesni diagrami, itd.
4. Programiranje (angl. programming): faza, ki prevede oblikovne specifikacije, proizvedene v fazi oblikovanja, v programsko kodo.
5. Namestitev (angl. installation): gre za zadnje korake pred končno implementacijo novega oziroma spremenjenega sistema, torej testiranje, uvajanje uporabnikov in prehod na nov sistem.
6. Izvajanje in vzdrževanje (angl. post-implementation): v tej fazi se preko uporabe in ocenjevanja nov sistem spreminja, da bi tako prišli do izboljšav oziroma izpolnitev novo nastalih zahtev.

Gre za drag, časovno potraten in nefleksibilen pristop. Poleg tega zahteva sodelovanje večjega števila uporabnikov in informatikov. V vseh fazah je razvoj skrbno načrtovan, izveden, nadzorovan in dokumentiran, z namenom minimalizirati možnost večjih napak.

2.1.2. ALTERNATIVNI PRISTOPI GRADNJE INFORMACIJSKIH SISTEMOV

2.1.2.1. Razvoj s prototipom

Pri razvoju informacijskega sistema s prototipom se uporabniku že v zelo zgodnjih fazah predstavi delujočo rešitev, ki pokriva le osnovne funkcije. Kasneje se s postopkom iteracije te funkcije dograjujejo in izpopolnjujejo, dokler ne pridemo do končne rešitve (Špehar, 2004, str. 9). Uporabniki preko interakcije s prototipom, ki je delujoča verzija informacijskega

sistema oziroma njegovega dela, lažje dobijo vpogled v lastne informacijske potrebe (Laudon, Laudon, 2000, str. 372).

Ključna prednost razvoja informacijskega sistema s prototipom je izboljšanje kvalitete začetnih faz razvoja, kar predstavlja osnovo za kvaliteten informacijski sistem. Omogoča možnost preizkušanja idej brez večjih stroškov, nizke razvojne stroške projekta, hiter razvoj začasne delujoče rešitve, učinkovito delitev dela med uporabniki in razvijalci, močno skrajšan čas razvoja sistema in promovira sodelovanje z uporabniki, ki so vpleteni v sam razvoj v že zelo zgodnjih fazah razvoja informacijskega sistema (Gradišar, Resinovič, 1996, str. 423). Ker uporabniki vedo, kakšen bo nov informacijski sistem, je lažja tudi sama faza končne implementacije (McLeod, Schell, 2001, str. 142).

Tega pristopa kljub svojim prednostim ne moremo uporabljati v vseh primerih. Dobro se obnese v časovnih stiskah, ko drugi pristopi ne bi omogočili pravočasne izdelave in kadar je možna delitev celotnega projekta na manjše, obvladljivejše in dokaj samostojne dele oziroma področja. Najbolje je uporabiti ta pristop v primeru, ko uporabnik ne ve točno, kaj dejansko želi imeti, ter kadar za uporabnika prijaznost programske rešitve predstavlja ključen dejavnik (Špehar, 2004, str. 14).

2.1.2.2. Razvoj s strani končnih uporabnikov

Organizacijske enote, ki se v organizaciji ukvarjajo z razvojem informacijskih sistemov, običajno ne morejo zadostiti vsem potrebam. Prihaja do zaostanka v razvoju uporabniških rešitev, zahteve uporabnikov se odlagajo ali pa celo zavračajo. Posledica opisanega stanja v organizacijah je pojav fenomena, imenovan razvoj s strani končnih uporabnikov (angl. End-user development). Pogoj za tak pristop je ustrezna osveščenost uporabnikov in določen nivo znanja s področja informatike, pa tudi zmogljivejša računalniška aparaturna oprema ter enostavna in do uporabnika prijazna programska oprema, kot so preglednice in jeziki četrte generacije (Gradišar, Resinovič, 1996, str. 427).

Brez dvoma ta pristop zagotavlja vrsto prednosti in koristi za organizacijo (Laudon, Laudon, 2000, str. 379):

- izboljšano določanje zahtev, saj uporabniki sami najbolj vedo, kaj za svoje delo potrebujejo,
- povečano sodelovanje in zadovoljstvo uporabnikov; ker uporabniki sami zasnujejo uporabniške rešitve, je večja verjetnost, da jih bodo tudi dejansko uporabljali,
- zmanjšana odvisnost od poklicnih informatikov v organizaciji; ki se tako lažje posvečajo večjim in bolj zahtevnim projektom,
- prihrani čas potreben za razvoj in stroške povezane s samim razvojem.

Seveda ima ta pristop tudi vrsto pomanjkljivosti (Špehar, 2004, str. 10, 11):

- slabo varovanje podatkov,
- neučinkovita raba resursov,
- neprimerno izobraževanje »razvijalcev«,
- neprimerna pomoč uporabnikom,
- nekonsistentni sistemi in
- podvajanje programskih rešitev.

2.1.2.3. Uporaba programskih paketov

Programski paket je vnaprej napisan, testiran in dokumentiran niz programov, ki so na voljo za najem oziroma nakup. Ni nujno, da gre le za programe, nekateri paketi namreč vključujejo tudi strojno opremo, na katerih se ti programi izvajajo.

V vsaki organizaciji obstajajo postopki in procesi, ki so splošni za tako rekoč vsako organizacijo, npr. obračun osebnih dohodkov, to pa predstavlja poslovni izziv za razvijalce programske opreme, da bi razvili sistem, ki bi ga lahko uporabljalo veliko organizacij. Nakup paketa uporabnikom močno skrajša čas uvedbe novega sistema in je običajno tudi ekonomsko upravičen. Ker je na trgu običajno več ponudnikov podobnih paketov, je pred nakupom potrebno zbrati čim več informacij o vseh in se odločiti za tistega, ki najbolj ustreza zahtevam. Pri tem je potrebno upoštevati lastnosti paketa tako z vsebinskega kot tehničnega vidika, boniteto prodajalca, prodajno ceno ter ceno svetovanja in vzdrževanja (Gradišar, Resinovič, 1996, str. 428).

Ključni problem pri tem pristopu h gradnji informacijskega sistema je v tem, da programski paketi praviloma ne pokrivajo vseh informacijskih potreb vsake organizacije. To dejstvo ima za posledico nakup večjega števila programskih paketov, kar povzroča dodatne stroške pri njihovem povezovanju, izobraževanju, ne samo končnih uporabnikov, ampak tudi lastnega oddelka za informatiko, in nenazadnje, ne izloča lastnega razvojnega dela na informacijskem sistemu.

Kot odgovor na našete pomanjkljivosti velika večina razvijalcev programskih paketov nudi osnovni paket, ki pa ga je pripravljena za ustrezno doplačilo modificirati oziroma prirediti specifičnim potrebam dotične organizacije in tako olajšati odločitev potencialnih uporabnikov za njihov proizvod.

Kljub vsem slabostim je programski paket verjetno narejen bolje, kot bi ga naredili sami, saj so ga izdelali specialisti, ki imajo izkušnje iz mnogih organizacij, ti paketi pa so tudi praviloma dobro dokumentirani in že testirani (Gradišar, Resinovič, 1996, str. 429). Zaradi vseh teh prednosti predstavljajo eno izmed najboljših opcij predvsem za manjše organizacije oziroma organizacije brez lastnega oddelka za informatiko.

2.1.2.4. Zunanje izvajanje (angl. Outsourcing)

Tega pristopa se poslužujejo organizacije, ki nimajo oziroma se upirajo uporabi lastnih, notranjih virov za izgradnjo informacijskega sistema. Najamejo zunanje partnerje, ki so specialisti na tem področju in nudijo tovrstne storitve. Ta pristop se je pojavil v zadnjem času predvsem kot posledica dejstva, da v primerjavi z vzdrževanjem lastnega računalniškega centra in osebja s področja informatike lahko predstavlja bistveno nižje stroške in manj organizacijskega napora. Ponudniki tovrstnih storitev pa zaradi izkoriščanja prednosti ekonomije obsega (isto znanje, spretnosti in zmogljivosti se delijo na več strank) lahko svoje storitve nudijo po konkurenčnih cenah (Laudon, Laudon, 2000, str. 381).

Kljub temu, da je vodenje projekta praviloma vedno znotraj organizacije, ima tudi ta pristop vrsto slabosti, ki jih mora vsaka organizacija pred odločitvijo za tak pristop upoštevati. V primeru, da za informacijski sistem skrbi neka druga organizacija, moramo upoštevati izgubo nadzora nad informacijsko funkcijo, sprejeti je potrebno karkoli nam ponudnik ponuja in za to plačati načeloma katerokoli ceno. Velik problem predstavljajo tudi poslovne skrivnosti in občutljive informacije, ki lahko končajo pri konkurentih (Laudon, Laudon, 2000, str. 381).

3. MANAGEMENT PODATKOV

Vsaka organizacija razpolaga z množico virov, s katerimi mora učinkovito upravljati za doseg zastavljenih ciljev. Kljub različni vlogi posameznega vira (človeški, finančni in materialni viri) je njihova skupna značilnost ta, da povzročajo stroške, hkrati pa organizaciji dajejo neko vrednost. Tako lahko tudi za podatke rečemo, da so vir organizacije, saj ravno tako povzročajo stroške in dajejo vrednost. Pravzaprav je vrednost podatkov edinstvena, saj je od njihove dostopnosti odvisna učinkovitost upravljanja z vsemi ostalimi viri organizacije. Zato je pomembno spoznanje, da so podatki za podjetje ključnega pomena in da se na njih gleda kot na vir, ki ga je potrebno upravljati učinkovito. Posledica spoznanja, da so podatki pomemben vir organizacije, je razvoj podatkovnih baz in sistemov za upravljanje podatkovnih baz (McFadden, Hoffer, 1988, str. 4).

Tradicionalno je bilo obdelovanje podatkov osredotočeno na posamezne organizacijske enote v organizaciji oziroma na obravnavo posameznih poslovnih funkcij. Zahteve uporabnikov so se reševale z razvojem posameznih programskih rešitev, vsak razvit uporabniški program pa je bil tako postopkovno orientiran in zasnovan na način, da je zadovoljil potrebe posameznega oddelka ali skupine uporabnikov. Ni obstajal model ali plan, na podlagi katerega bi potekal razvoj uporabniških rešitev, kar je imelo za posledico, da je bila vsaka rešitev zasnovana s svojimi datotekami podatkov. Velika večina podatkov v teh novih datotekah je že bila v obstoječih datotekah, ki so jih obravnavali drugi uporabniški programi. Glavni razlog kreiranja novih datotek je bil v tem, da je bilo potrebno za nove potrebe obstoječe datoteke spremeniti oziroma spremeniti njihovo strukturo. Sprememba strukture datotek pa je

neposredno vplivala na uporabniške programe, ki jih je bilo potrebno spremeniti ali v posameznih primerih napisati na novo, kar pa je bilo velikokrat bolj tvegano in dražje od kreiranja novih podatkovnih datotek (Vožič, 1991, str. 15). Vse to je pripeljalo do vrste slabosti (McFadden, Hoffer, 1988, str. 11-15):

- nekontrolirano podvajanje podatkov,
- nekonsistentnost podatkov,
- neprilagodljivost podatkov,
- omejena možnost delitve podatkov,
- majhna možnost uvajanja standardov,
- nizka produktivnost programerjev,
- velika potreba po vzdrževanju uporabniških programov.

Zaradi zgoraj naštetih slabosti je prišlo do neizogibnega spoznanja, da je potreben sistematičen in standardiziran pristop, ki ga imenujemo podatkovni pristop oziroma pristop, ki temelji na integriranih podatkovnih virih (angl. data base approach). Ta sloni na temeljiti analizi današnjih in bodočih informacijskih potreb ter načrtu izgradnje ustrezne podatkovne baze, ki je vključen v plan izgradnje računalniško podprtega informacijskega sistema, ta pa seveda predstavlja bistveni element strateških planov organizacije. Prednosti v primerjavi s tradicionalnim pristopom so minimalno podvajanje podatkov, konsistentnost podatkov, integriteta podatkov, deljivost podatkov, uveljavljanje standardov, poenostavljeno razvijanje uporabniških rešitev, poenotena zaščita podatkov, zasebnost podatkov, podatkovna neodvisnost in zmanjšana potreba po vzdrževanju uporabniških programov (Vožič, 1991, str. 18).

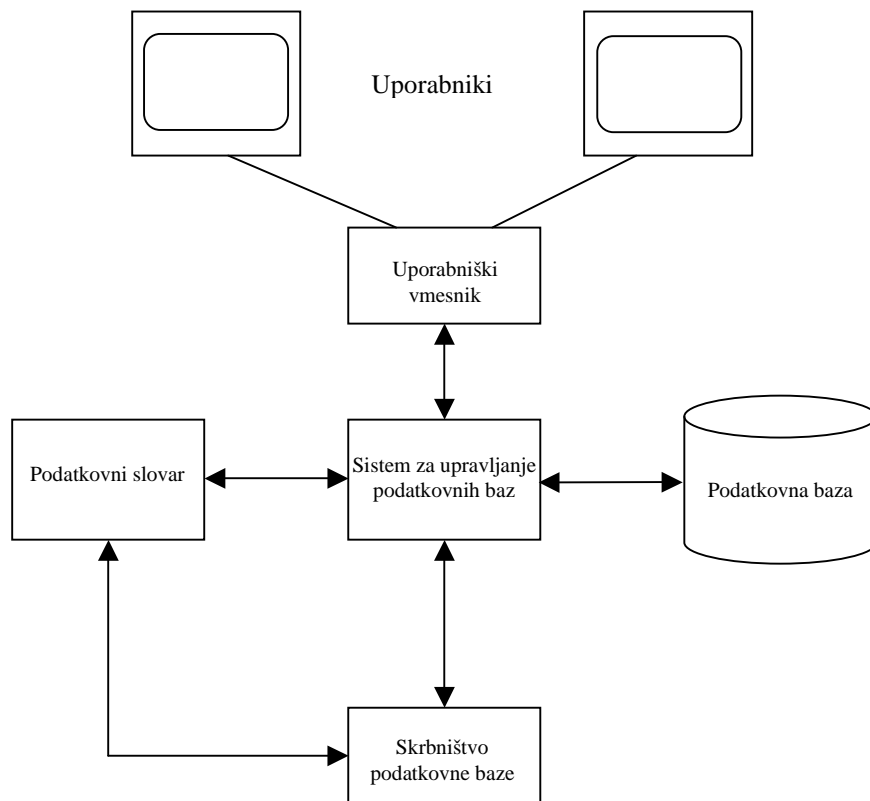
3.1. KOMPONENTE OKOLJA PODATKOVNIH BAZ

Glavne komponente tipičnega okolja podatkovnih baz sestajajo iz šestih komponent, kot je prikazano na Sliki 1 (na strani 8). S proučevanjem teh komponent in njihovih medsebojnih povezav lažje razumemo pristop, ki temelji na integriranih podatkovnih virih, in njegove prednosti.

3.1.1. UPORABNIKI

Uporabniki so vsi, ki imajo na tak ali drugačen način opravka s podatki organizacije, od uporabnikov na operativnem nivoju, ki imajo opravka z vsakodnevnimi transakcijami, pa do managerjev na taktičnem in strateškem nivoju. Na splošno lahko govorimo o treh osnovnih kategorijah interakcij uporabnikov s podatki podatkovne baze organizacije: samo branje (angl. read-only), dodaj/briši (angl. add/delete) in sprememba (angl. modify). Vse te zahteve obdela sistem za upravljanje podatkovnih baz (SUPB).

Slika 1: Komponente okolja podatkovnih baz



Vir: McFadden, Hoffer, 1988, str. 23.

3.1.2. UPORABNIŠKI VMESNIK

Uporabniški vmesnik je sestavljen iz jezikov in drugih značilnosti, s pomočjo katerih so uporabniki v interakciji s podatkovno bazo. Preko teh vmesnikov so kanalizirane vse zahteve po podatkih in vse transakcije z njimi. Različni uporabniki imajo potrebo po različnih vmesnikih. Programer na primer potrebuje proceduralen jezik, kot so C, Pascal, Delphi in Visual Basic, managerju pa bo verjetno udobneje v primeru vmesnika, zasnovanega s pomočjo menijev. Zaradi napredka v razvoju programske opreme so ti uporabniški vmesniki postali zelo prijazni uporabniku, saj je poleg sistema prej omenjenih menijev do velikega napredka prišlo tudi v sistemih prepoznavanja glasu, uporabi miške in podobno.

3.1.3. PODATKOVNA BAZA

Podatkovna baza (angl. database) je urejena zbirka medsebojno povezanih podatkov, ki je shranjena na računalniškem nosilcu podatkov. S podatki, ki so shranjeni v podatkovni bazi, je opisan del realnega sveta, kar pomeni, da se morajo spremembe v realnem svetu odražati v podatkovni bazi. Podatkovna baza mora biti integrirana, kar pomeni, da vsebuje podatke za

mnoge uporabnike, pri čemer vsakega od njih praviloma zanima le manjši del celote. Ker pa se potrebe uporabnikov med seboj prekrivajo, pomeni, da z integrirano podatkovno bazo zmanjšamo podvajanje podatkov na najmanjšo možno mero (Kovačič et al., 2004, str. 116).

Glede na način uporabnikovega dostopa do podatkovne baze lahko razlikujemo (Kovačič et al., 2004, str. 117, 118):

1. Dostop z vnaprej pripravljenimi programskimi rešitvami (zaslonske maske za vnos podatkov, pripravljena poročila za izpis na tiskalnik itd); tak način je primeren za aktivnosti, kjer so postopki vnaprej dobro definirani, praviloma pri izvajanju procesov na operativni ravni.
2. Dostop neposredno prek sistema za upravljanje podatkovnih baz (običajno ad hoc poizvedbe); tak način pride v poštev pri opravilih, kjer nismo vnaprej predvideli posebnih programov, ker pri načrtovanju informacijskega sistema teh potreb nismo poznali ali pa se pojavijo tako redko, da izdelava programov ne bi bila smotrna. To je značilno predvsem za odločitvene aktivnosti na višjih ravneh odločanja (taktičnem in strateškem).

Glede na vrsto opravila pa lahko uporabo podatkovne baze razdelimo na (Kovačič et al., 2004, str. 118):

1. vnos novih podatkov,
2. spreminjanje (ažuriranje) obstoječih podatkov,
3. brisanje podatkov,
4. poizvedovanje.

Pri izvajanju poslovnih procesov na operativni ravni so običajna opravila vseh vrst, na višjih ravneh managementa pa je najbolj običajno opravilo poizvedovanje, pri čemer nas v nasprotju z operativno ravno le redko zanima posamezna entiteta, pač pa večinoma uporabljamo sumarne podatke o več entitetah oziroma poslovnih dogodkih.

Zaradi dejstva, da imajo uporabniki na različnih ravneh organizacije različne potrebe po podatkih in različne tipe interakcij s samo podatkovno bazo, se tudi zahtevane zmogljivosti podatkovnih baz razlikujejo glede na različne tipe uporabnikov. V transakcijskem okolju se zahteva predvsem hiter dostop do podatkov oziroma hitro izvajanje transakcij ob velikem številu hkratnih uporabnikov. Zahteva se zagotavljanje celovitosti podatkov, varnost pred napakami na strojni in programski opremi ter čim hitrejši povratek v stabilno stanje v primeru napak. Standard za podatkovne baze pa sta prav tako lastnost možnosti distribuiranih podatkovnih baz in lastnost vzporednega procesiranja za večprocesorske računalnike. Na analitičnem področju, kjer se srečujemo s podatkovnimi skladišči, so predvsem pomembne zahteve vezane na upravljanje z velikimi količinami podatkov. Za izvajanje operacij imamo v kvalitetnih bazah posebne mehanizme, kot so particioniranje velikih tabel, uporaba različnih indeksnih struktur, optimizacija izvajanja povpraševanj glede na naravo povpraševanj v podatkovnih skladiščih, hiter prenos podatkov iz transakcijskega v analitično okolje in podobno (Komel, 1998, str. 19, 20).

3.1.4. PODATKOVNI SLOVAR

Podatkovni slovar je eno od zelo pomembnih orodij, ki ga uporabljamo pri procesu planiranja, zasnove in vzdrževanja podatkovne baze. Uporaba tega orodja je še posebej pomembna pri vzdrževanju dokumentacije, snovanju podatkovne baze, generiranju meta podatkov (podatkov o podatkih) in nadziranju konsistentnosti.

Pri procesu planiranja podatkovne baze je slovar podatkov uporabljen za dokumentiranje podatkov o poslovnih funkcijah, procesih, aktivnostih in entitetah organizacije. Glavni cilj tega procesa je razvoj globalnega modela podatkov, pri čemer nam slovar podatkov služi za opis podatkov, entitet ter povezav med njimi. Podatki, ki nas v tem procesu zanimajo so (Laudon, Laudon, 1991, str. 819):

- kateri podatki so na voljo,
- kje se podatki nahajajo,
- katere attribute vsebujejo,
- kdo je odgovoren za posamezne podatke (lahko je posameznik ali pa poslovna funkcija, kot npr. proizvodnja, trženje),
- kako so podatki uporabljeni,
- kdo ali katera funkcija ima dostop do posameznih podatkov,
- kdo ali katera funkcija ima pravice spreminjanja podatkov,
- kako so podatki med seboj povezani,
- varnostne in druge omejitve.

Zasnova baze podatkov sestoji iz procesa konceptualne, logične in fizične zasnove. Slovar podatkov, kot orodje zasnove, se pri tem uporablja za dokumentiranje logične in fizične zasnove ter za generiranje uporabniških priročnikov in drugih oblik dokumentacije. V procesu implementacije, ko podatkovno bazo prvič napolnimo s podatki ter jo testiramo, razvijamo tudi uporabniške programe. Slovar podatkov nam pri tem služi za generiranje informacij za najrazličnejše uporabnike. Za razvijalce uporabniških rešitev so to informacije o podatkovnih elementih, tipih zapisov in ostalih meta podatkih, potrebnih za razvoj uporabniških rešitev. Skrbnik podatkovne baze uporablja pri polnjenju podatkovne baze slovar podatkov za razreševanje nekonsistentnosti, kot so sinonimi in homonimi ali neustrezni tipi podatkov. Poleg tega lahko slovar uporabljamo tudi za generiranje testnih podatkov za testiranje najrazličnejših komponent sistema. Ko sistem uvedemo in je v uporabi, nastopi faza vzdrževanja podatkovne baze. V tej fazi skrbnik baze podatkov potrebuje slovar podatkov za kontrolo sistema pri izvajanju zaščite podatkov ter za vzdrževanje integritete podatkovnih baz (Vožič, 1991, str. 24).

Podatkovni slovar vsebuje standardno definiranost vsakega podatkovnega elementa, ki ga organizacija želi v svojem katalogu podatkov. Posamezen podatkovni element je definiran le enkrat, s čimer se izognemo nekonsistentnosti in podvajanju podatkov. Uporabiti je torej

potrebno standardne obrazce, ki opisujejo podatke, z namenom zagotoviti enoličnost definicij podatkovnih elementov podatkovnega slovarja (McFadden, Hoffer, 1988, str. 245).

Podatkovni slovar omogoča učinkovito upravljanje in kontrolo podatkovnih virov tudi z drugih vidikov (Laudon, Laudon, 1991, str. 821):

- *Informiranje o razpoložljivosti podatkov*: uporabnikom informacijskega sistema nudi informacije o tem, kateri podatek je na voljo, kaj pomeni, kje se ga da dobiti, ali je dostopen brez omejitev, itd.
- *Lažja kontrola nad spremembami informacijskega sistema in podatkovnih virov*: s pomočjo podatkovnega slovarja lahko natančno ugotovimo, katere dokumente, programe, transakcije, poročila ali sisteme moramo spremeniti v primeru spremembe v velikosti, formata ali atributa posameznega podatkovnega elementa.
- *Učinkovitejši razvoj sistema*: nudi pregleden seznam obstoječih podatkovnih standardov, ki ga potrebujejo sistemski analitiki in programerji, kar olajša njihovo delo.
- *Zagotavljanje varnosti in kontrole*: prikazuje pooblastila posameznega uporabnika/poslovne funkcije za dostopnost ali spreminjanje podatkov ter kdo je odgovoren za posamezen podatek.

3.1.5. SKRBNIŠTVO PODATKOVNE BAZE

Potreba po upravljanju podatkov organizacije je narekovala uvedbo nove poslovne funkcije, in sicer tako imenovano funkcijo skrbništva nad podatkovnimi bazami (angl. Data Base Administration, DBA). Organiziranje in izvajanje te funkcije je bistvenega pomena za uspešno uvedbo podatkovnega pristopa. Podatkovna baza pripada celotni organizaciji in ni last posamezne organizacijske enote ali posameznika. Skrbnik podatkovne baze, ki izvaja funkcijo skrbništva nad bazami podatkov, je skrbnik podatkov organizacije, kot je finančnik skrbnik nad finančnimi sredstvi organizacije. Skrbnik podatkov mora razviti postopke za zaščito in kontrolo nad podatki ter razreševati morebitne spore, ki nastajajo pri centralizaciji podatkov, ko jih uporabljajo različni uporabniki v organizaciji (Vožič, 1991, str. 138).

Naloge skrbnika podatkovne baze so poleg že naštetih še skrb za dostopnost, odzivnost in varnost podatkovne baze, izdelava zahtevnejših ad hoc poizvedb, izobraževanje in pomoč uporabnikom in izbor sistema za upravljanja podatkovnih baz (Kovačič et al., 2004, str. 127).

Izbor skrbnika podatkovne baze in organiziranje funkcije skrbništva nad podatkovnimi bazami je zelo odgovorna naloga, ki jo je potrebno izvesti z veliko mero skrbnosti. Skrbnik podatkovne baze mora biti sposoben vodenja in razreševanja spornih situacij, ki se normalno pojavljajo, ko pride do pomembnih sprememb v organizaciji. Odgovoren mora biti predvsem vodstvu podjetja ter vodjem posameznih organizacijskih enot in ne posameznim računalniškim strokovnjakom (McFadden, Hoffer, 1988, str. 339).

3.1.6. SISTEM ZA UPRAVLJANJE PODATKOVNIH BAZ

Značilnost uporabe podatkovnih baz je predvsem ta, da uporabnik (ali program) ne more neposredno dosežati ali spreminjati podatkov v bazi. S tem je dosežena neodvisnost med programi in podatki, ki jo omogočajo prav SUPB. SUPB posreduje zahtevo, ta pa to izvede, kar pomeni, da spremeni podatke ali poišče in vrne odgovor na postavljeno izvedbo. Ta neodvisnost je dosežena s podatkovnim slovarjem, v katerem je shranjen opis podatkovne baze; njene strukture, tipov podatkov, pomena podatkov itd.

Sistemi za upravljanje podatkovnih baz (SUPB, angl. Database Management System, DBMS) so zbirka programov, ki omogočajo tvorbo, uporabo in vzdrževanje podatkovnih baz. Z njimi torej izvajamo vse aktivnosti, povezane s podatkovno bazo, razen načrtovanja, kateremu so namenjena programska orodja CASE (angl. Computer-Aided Software Engineering). To pomeni, da so SUPB namenjeni raznolikim uporabnikom, od načrtovalca podatkovne baze, programerja, skrbnika, do uporabnika (Kovačič et al., 2004, str.121).

SUPB omogočajo tudi vsa opravila, povezana z uporabo podatkovne baze (vnos, spreminjanje, brisanje in poizvedovanje). Eden najpogosteje uporabljenih jezikov za delo s podatkovnimi bazami je SQL (angl. Structured Query Language), ki je standard za delo z relacijskimi podatkovnimi bazami (Kovačič et al., 2004, str. 123).

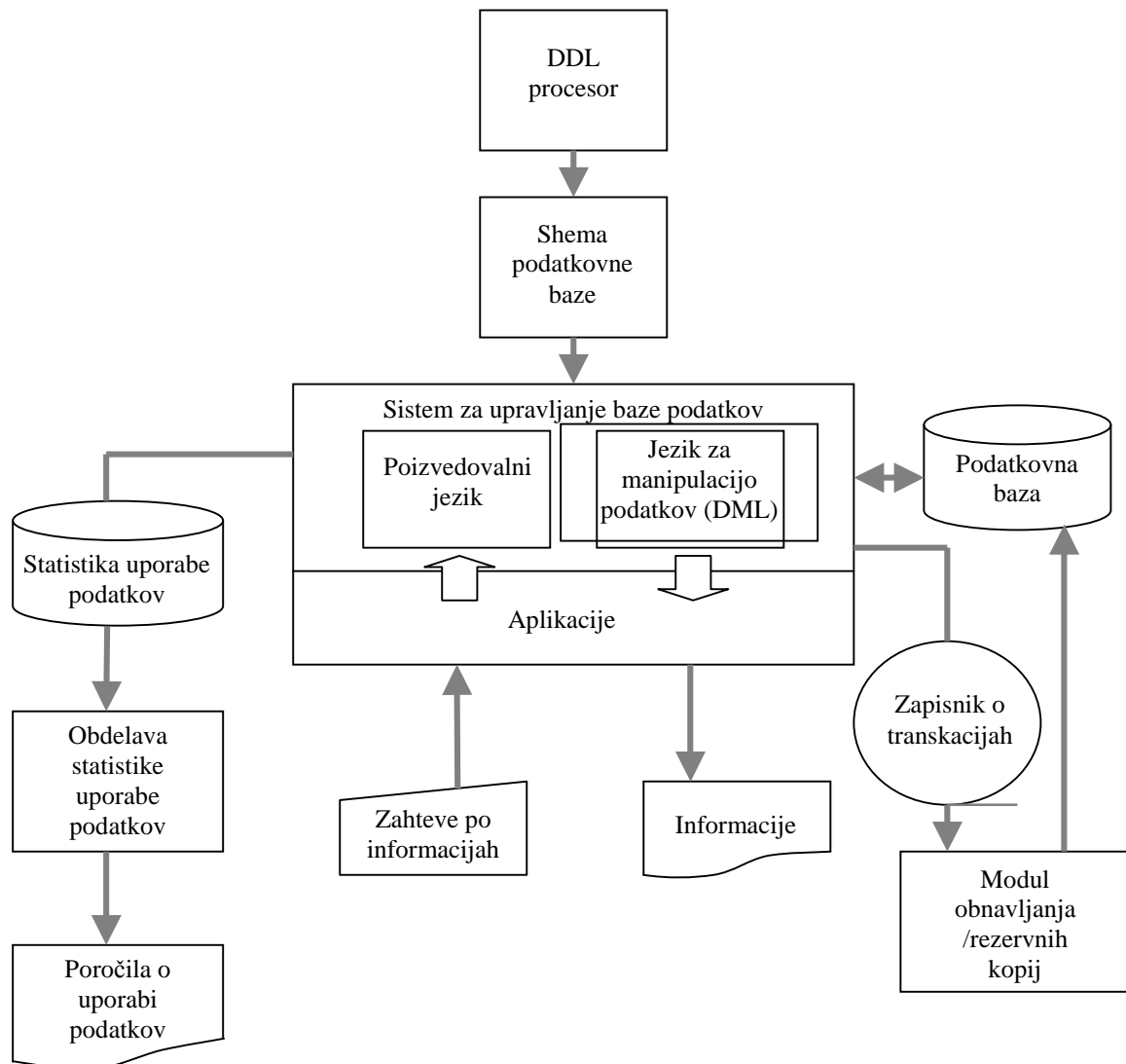
Slika 2 (na strani 13) prikazuje glavne elemente sistema za upravljanje podatkovnih baz. Ko imamo enkrat narejen podatkovni slovar, ga s pomočjo jezika za opisovanje podatkov (angl. Data Description Language, DDL), ki je del vsakega SUPB, vnesemo v sam SUPB. Rezultat tega procesa je shema podatkovne baze, ki običajno določa karakteristike, kot so: ime podatkovnega polja, sinonime, tip podatka, število pozicij, različna pravila celovitosti podatkov in podobno. Vsi sistemi za upravljanje podatkovnih baz imajo torej vsebovan DDL, s pomočjo katerega prevajajo uporabnikov opis strukture podatkovne baze v format, ki ga lahko uporablja SUPB.

Modul obdelave statističnih podatkov o uporabi podatkov podatkovne baze je potreben za identifikacijo podatkov, ki so v uporabi, kdo jih uporablja, kdaj so uporabljeni in tako naprej. Te informacije so namreč izjemnega pomena, kadar bi radi s pomočjo izdelave profila uporabe podatkovne baze ugotovili, kako lahko s spreminjanjem same strukture podatkovne baze naslavljamo potencialno procesno neučinkovitost. S pomočjo teh podatkov SUPB omogoča tudi zaščito pred nepooblaščenim dostopom tako, da ko skrbnik podatkovne baze vsakemu uporabniku dodeli uporabniško ime, na katero so vezane pravice dostopa (vnos, brisanje itd.), SUPB te upošteva in preverja.

Aktivnost zapisovanja in spremljanja transakcij (angl. transaction log), torej vseh sprememb, do katerih pride zaradi vsakodnevnih interakcij uporabnikov s podatkovno bazo, je nujna za delovanje modula izdelovanja rezervnih kopij, zrcaljenja (angl. Mirroring) in obnavljanja

(angl. Backup/Recovery modul). V primeru, da pride do napak ali poškodb same podatkovne baze, je s pomočjo tega modula omogočena vzpostavitev stanja pred nastankom neljubega dogodka.

Slika 2: Model sistema za upravljanje podatkovnih baz



Vir: McLeod, Schell, 2001, str. 190.

Element sistem za upravljanje baz podatkov je najpomembnejši v predstavljenem modelu, saj obdeluje uporabnikove zahteve po podatkih. Nadzoruje poizvedovalni jezik in jezik za manipulacijo s podatki (ang. Data Manipulation Language, DML) in je odgovoren za izdelavo statistike uporabe podatkov, ki je procesiran s strani modula za obdelavo statističnih podatkov uporabe podatkov, in za zapisnik o transakcijah, ki so vhodni element modula izdelovanja

rezervnih kopij in obnavljanja. Ta element je vsebovan v vsakem sistemu za upravljanje podatkovnih baz.

V okviru SUPB je del programov oziroma del programa namenjen tudi aktivnostim, povezanih z vzdrževanjem oziroma skrbništvom podatkovne baze. Omogočajo nadzor celovitosti podatkov (angl. Data Integrity). Gre za semantične (pomenske) omejitve, ki jih lahko načrtovalec definira na podatkovni bazi, te omejitve pa morajo izvirati iz realnega sveta in gre dejansko le za predstavitev poslovnih pravil, z upoštevanjem teh pravil pa SUPB uporabniku onemogoča, da bi nenamerno privedel podatkovno bazo v nedovoljeno stanje. V grobem lahko omejitve celovitosti delimo na tri vrste (Kovačič et al., 2004, str. 123-125):

1. omejitve območja (določajo, kakšne vrednosti lahko zavzame atribut),
2. referenčna celovitost (določa, da lahko neki atribut zavzame le vrednosti, ki jih ima ustrezní ključ),
3. eksplicitna celovitost (pogoji in dovoljene vrednosti oziroma zahteve so navadno definirane s konkretnimi vrednostmi).

Izbira SUBP je večkriterijska odločitev, kjer moramo poleg stroškovnih vidikov upoštevati predvsem (Kovačič et al., 2004, str. 137):

1. povezljivost z obstoječimi sistemi,
2. ali so funkcionalnosti, ki jih sistem ponuja, skladne s potrebami,
3. zmogljivost sistema in nadgradljivost v primeru povečanja obsega poslovanja,
4. podporo ponudnika in možnosti izobraževanja.

Pri tem se ne smemo omejevati zgolj na stroške nakupa sistema, pač pa moramo upoštevati celotne stroške lastništva (angl. Total Cost of Ownership, TCO), kar vključuje tudi stroške vzdrževanja, nadgradenj, podpore, izobraževanja, prehoda in drugo.

4. PODATKOVNO MODELIRANJE

Modeli se uporabljajo na različnih področjih in na različne načine. Teorija, ki opisuje neki proces, je model tega procesa, oziroma povedano drugače, opis nekega dogodka lahko že štejemo za model tega dogodka. Pri tem velja, da morata imeti original in model tega originala enake strukture. Pod strukturo sistema razumemo izbrani niz odnosov med deli, na katere smo sistem razdelili. Razdelitev sistema na dele in izbor odnosov med njimi je razlog, zaradi katerega sploh definiramo strukturo. Na istem sistemu lahko definiramo nekaj različnih struktur in skladno s tem razvijemo nekaj med sabo popolnoma različnih modelov tega sistema (Vožič, 1990, str. 43).

Osnovni mehanizem modeliranja je torej abstrakcija, tj. posplošenje, odmišljanje nebitvenega oz. manj pomembnega. Z drugimi besedami, realnost želimo predstaviti tako, da izpostavimo nekatere pomembne vidike problema, medtem ko ostale zanemarimo.

4.1. ANALIZA PODATKOVNIH POTREB

Osnovno izhodišče pri modeliranju podatkov predstavljajo modeli poslovnih procesov, saj je v modelih prikazano, kaj v proces ter posamezne aktivnosti vstopa in kaj iz procesa in posameznih aktivnosti izstopa. Za zagotavljanje kakovosti podatkov in nemotenega informacijskega toka morajo biti podatki enovito opredeljeni skozi celoten proces in na ravni organizacije. Poleg modelov procesov za potrebe podatkovnega modeliranja oz. analize podatkovnih potreb uporabljamo še druge tehnike analize, predvsem (Kovačič et al., 2004, str. 99):

- zbiranje obstoječe dokumentacije,
- pogovore z uporabniki (intervjuji),
- analiza obstoječih podatkovnih virov oz. dokumentacije.

Te aktivnosti praviloma izvajamo vzporedno z analizo poslovnih procesov, v primeru podrobnejšega podatkovnega modeliranja za potrebe načrtovanja podatkovnih baz pa je potrebno izvesti še dodatno analizo. Analiza potreb podatkov za izvajanje operativnih procesov pa ne sme zanemariti dejstva, da je pri razvoju operativnih baz podatkov potrebna tudi analiza potreb po podatkih za poslovno odločanje na strateški in taktični ravni, saj moramo zadostiti vsem informacijskim potrebam vseh uporabnikov informacijskega sistema, ki bo zgrajen s pomočjo ugotovitev teh analiz.

V splošnem v fazi analize ugotavljamo, kateri so ključni pojmi poslovanja oziroma o katerih stvareh (objektih, dogodkih itd.) nameravamo hraniti podatke v podatkovnem viru, kako so te stvari med seboj povezane, katere podatke o teh rečeh želimo hraniti v podatkovni bazi, kako so ti podatki med seboj povezani, kar vpliva na organizacijo podatkov in morebitna poslovna pravila, ki vežejo oziroma omejujejo podatke.

Rezultat uporabe tehnik podatkovne analize večinoma povzroča težave pri nadaljnjem načrtovanju, saj pogosto ne pokriva vseh potreb po podatkih, mnogokrat opisi niso pomembni za izdelavo podatkovnega vira, so dvoumni ali pa si opisi različnih uporabnikov med seboj nasprotujejo (neenotno razumevanje poslovnih konceptov, pravil itd.). Zato izdelamo formalni zapis ugotovitev (pomanjkljivosti, protislovja, dvoumnosti). Z vidika komunikacije pa je pomembno poskrbeti, da je uporabljena tehnika enostavna in razumljiva tudi neinformatiku, hkrati pa dovolj formalna, da preprečuje dvoumne prikaze. Zato za (poslovno) podatkovno modeliranje uporabljamo večinoma grafične tehnike modeliranja (Kovačič et al., 2004, str. 101).

4.2. MODELI PODATKOV

S podatkovnim modelom prikažemo podatkovne oz. informacijske potrebe organizacije. Prikazuje objekte, dogodke, aktivnosti delovanja, podatke o teh, ki jih potrebujemo, povezave med objekti in nekatera (poslovna) pravila. Izdelamo jih lahko z različnih vidikov: od povsem poslovnih modelov, ki so tehnološko povsem neodvisni in opredeljujejo elemente modela z vidika poslovanja, do načrtov baz podatkov, ki so vezani na izbrano tehnologijo. Prav tako lahko izdelamo podatkovne modele na različnih ravneh. Praviloma začnemo z izdelavo globalnega podatkovnega modela na ravni celotnega podjetja, ki opredeljuje ključne pojme poslovanja, vendar pri tem ni zelo podroben. V nadaljevanju pa v okviru informatizacije procesov modele podrobneje razdelamo, skladno s podatkovnimi potrebami procesa, do zelo podrobnih modelov v okviru razvoja posamezne programske rešitve. Ker globalni podatkovni model predstavlja osnovno izhodišče razvoja baz podatkov, mora ta biti enostaven, razumljiv in zato praviloma predstavljen v grafični obliki. S stališča standardizacije podatkov podjetja oziroma njihove kakovosti je nedvomno pomemben pristop oziroma postopek njihovega modeliranja. V tem procesu je s stališča razumevanja informacijskih potreb pomembna izbira predstavitvene tehnike, s stališča zasnove, uporabe in vzdrževanja podatkovne baze pa izbira ustreznih informacijskih orodij (SUBP, CASE itd.) (Kovačič et al., 2004, str. 98, 99).

4.2.1. TRINIVOJSKA ARHITEKTURA

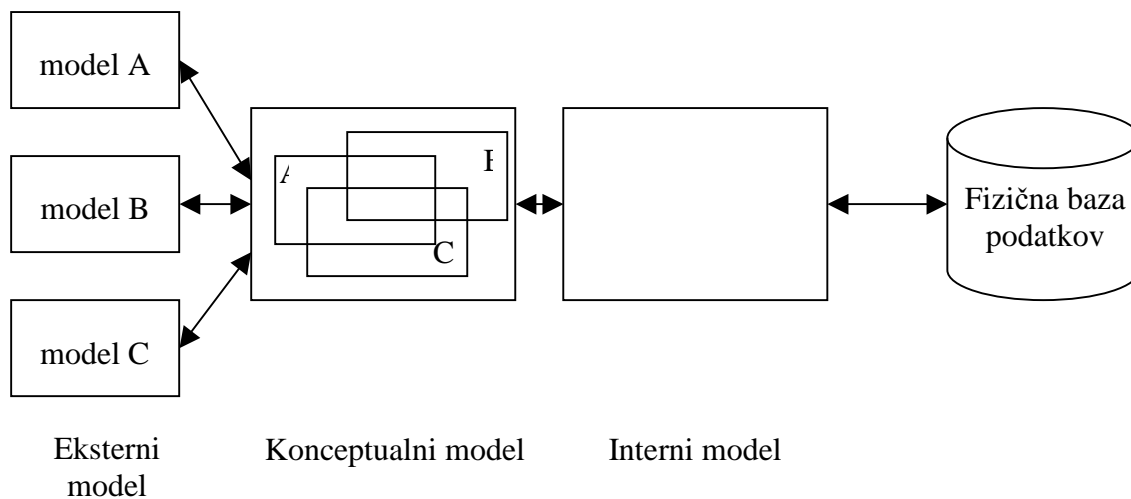
Leta 1975 je študijska skupina ANSI/X3/SPARC razvila metodo za izdelavo modelov podatkov in podatkovnih slovarjev, ki je poimenovana model ANSI/SPARC. Leta 1985 je metodo sprejela tudi mednarodna organizacija za standardizacijo ISO in jo poimenovala »Konceptualna shema in model trinivojske arhitekture« (Tkalac, 1988, str. 5).

Baza podatkov omogoča dostop do podatkov najrazličnejšim uporabnikom v organizaciji, vsak izmed njih pa ima na te podatke specifične poglede, saj ti izhajajo iz značilnosti posameznega poslovnega procesa. Kot je razvidno iz Slike 3 (na strani 17), model trinivojske arhitekture omogoča pogled na podatke s treh nivojev abstrakcije, njen cilj pa je ustvariti takšne uporabniške poglede, da bodo prirejeni vsakemu uporabniku podatkovne baze:

- *Zunanji model* (angl. external data model): predstavlja poglede uporabnikov kot podmnožice konceptualnega modela, odvisne od specifičnih potreb.
- *Konceptualni model* (angl. conceptual data model): definira celotne informacijske vire na abstraktnem nivoju, vključuje vse entitete in njihove medsebojne povezave, omejitve dostopa ter je neodvisen od fizične implementacije.
- *Notranji model* (angl. internal data model): predstavlja definicijo fizične implementacije podatkovne baze v izbranem sistemu za upravljanje podatkovnih baz in je tako edini odvisen od uporabljene tehnologije.

Najnižji nivo, ki predstavlja fizično bazo podatkov, dejansko obstaja, vsi drugi so le različne stopnje abstrakcije.

Slika 3: Model trinivojske arhitekture



Vir: Tkalac, 1998, str. 6.

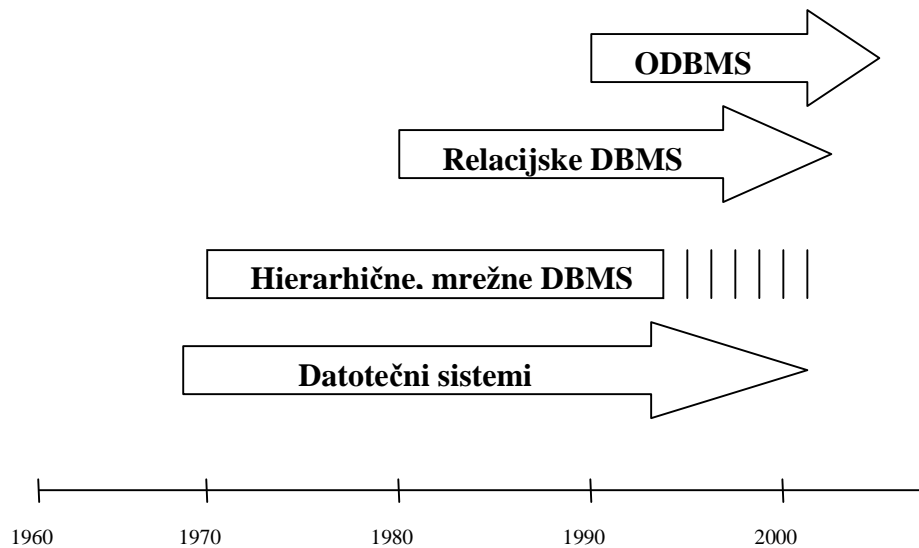
S takšno organizacijo trinivojske arhitekture je dosežena podatkovna neodvisnost, kar naj bi pomenilo minimalen vpliv sprememb na obstoječe uporabnike. Ločimo dve vrsti neodvisnosti. Fizično neodvisnost podatkov, ki omogoča modifikacije notranjega modela z izbiro drugačne fizične organizacije, pristopnih metod in podobno, ter logično neodvisnost podatkov, ki omogoča modifikacijo konceptualnega modela z dodajanjem novih entitet ali atributov za razvoj podatkovne baze, s tem da ostanejo obstoječi uporabniški pogledi nespremenjeni (Grad, Kovačič, Barle, 1990, str. 26).

Notranji modeli, ki so neposredno odvisni od izbire tehnologije fizične implementacije, torej izbire SUPB, so zato vedno predstavljeni v hierarhični, mrežni, relacijski, objektni ali kateri drugi obliki, pač odvisno od izbire SUPB, in so kot taki dokaj učinkovit način predstavitve podatkovnih struktur. Za prikaz konceptualnega in zunanega modela pa je bolje uporabiti od notranje implementacije neodvisne oblike in načine, kar naj po eni strani omogoča učinkovitejši prikaz podatkovnih struktur na logični ravni ter po drugi strani omogoči razvoj od tehnologije implementacije neodvisnih programskih orodij za razvoj modelov. Najpomembnejši tehniki takšnega prikazovanja sta model entitet-povezav in semantični model.

4.2.2. MODELI PODATKOV ODVISNI OD UPORABLJENE TEHNOLOGIJE

Slika 4 (na strani 18) nam prikazuje evolucijo sistemov za upravljanje podatkovnih baz, vse od enostavnih datotečnih sistemov koncem 60-ih, pa vse do pojava komercialnih proizvodov objektnih podatkovnih baz koncem 80-ih in začetku 90-ih let.

Slika 4: Prikaz generacij sistemov za upravljanje s podatki



Vir: Domajnko, Heričko, Rozman, 1997, str. 184.

4.2.2.1. Starejši podatkovni modeli

Datotečni podatkovni modeli

Predstavljajo podatkovne objekte v obliki zapisov, grupiranih v datoteke. Možnosti povezovanja objektov oziroma vzpostavljanja povezav med njimi so zelo skromne (indeksne in invertirane tabele). Čeprav ti modeli predstavljajo začetno stopnjo modeliranja podatkov, še danes velik del računalniške obdelave podatkov temelji na njih (Vintar, 1990, str. 48)

Hieararhičen model podatkov

S hierarhičnim modelom lahko zelo dobro opisujemo marsikatero stanje ali dogajanje in se zaradi tega v praksi pojavljajo dokaj pogosto. Gre za naravno strukturo mnogih baz podatkov. Učinkoviti so zlasti pri tistih aplikacijah, ki jih ponavljajoče uporabljajo vedno na isti, vnaprej predviden način.

Datoteke so povezane v obliki drevesa. Vsak zapis mora imeti enega lastnika (angl. parent), ki se hierarhično nahaja tik nad njim. Izjema so zapisi, ki se nahajajo prav na vrhu in zato nimajo lastnika. Lastniki ali starši pa imajo lahko več otrok (angl. child). Gre torej za razmerje eden proti mnogim (1:N). To pomeni, da mora pri dostopu do podatkov v hierarhičnem modelu podatkov uporabnik slediti hierarhiji. Začeti je potrebno pri stebelu drevesa oziroma pri prvem zapisu v datoteki, saj so zapisi v datotekah povezani v verigo. Vsak zapis, razen zadnjega, pa

ima kazalec s fizičnim naslovom zadnjega zapisa. S pomočjo kazalcev se tako sprehajamo po verigi in iščemo želeni zapis (Gradišar, Resinovič, 1996, str. 290-292).

Mrežni model podatkov

Nekatere podatke je bilo veliko bolj naravno modelirati z več kot enim lastnikom na enega otroka, torej s tako imenovano M:N povezavo. Leta 1971 na konferenci o jezikih podatkovnih sistemov ([Conference on Data Systems Languages - CODASYL](#)) formalno definirajo mrežni model podatkov (Database Models, 2005).

Mrežni model podatkov je variacija hierarhičnega modela, baze podatkov so lahko prevedene iz enega v drugega in obratno, da bi se na ta način optimizirala procesna hitrost in sama prikladnost (Laudon, Laudon, 2000, str. 238). Tako kot hierarhičen je tudi mrežni model podatkov primeren za opisovanje mnogih praktičnih situacij. Vendar pa je njegova uporaba povezana s številnimi podobnimi problemi kot uporaba hierarhičnega modela. Z vidika programerja ostaja problem obsežnosti in zapletenosti, z vidika končnega uporabnika pa osnovni problem nepraktičnosti pri izražanju novih poizvedb, ki jih terjajo nove situacije (Resinovič, Gradišar, 1996, str. 293).

Ker so zapisi povezani po načelu mnogi z mnogimi, vsak zapis ima torej lahko več lastnikov hkrati, pri mrežni strukturi ni potreben dostop do posameznega zapisa preko nadrejenih datotek (Resinovič, Gradišar, 1993, str. 202). Mrežne strukture zmanjšujejo podvajanje in odvečnost podatkov ter se v določenih situacijah (predvsem ko so vpletene M:N povezave) odzivajo hitreje. Vendar pa ima to svojo ceno, saj se število kazalcev v mrežnih strukturah zelo hitro večja in s tem potencialno otežuje vzdrževanje in samo delovanje (Laudon, Laudon, 2000, str. 238).

4.2.2.2. Relacijski model podatkov

Relacijski model podatkov temelji na matematični teoriji relacij. Podatki so organizirani v relacijske podatkovne strukture, ki pa so predstavljene z dvodimenzionalnimi tabelami. Praviloma hranimo v eni tabeli podatke o vseh entitetah nekega entitetnega tipa ali o vseh povezavah nekega tipa povezav. Vsaka vrstica predstavlja eno entiteto ali povezavo, vsak stolpec pa predstavlja en atribut, kjer so shranjene vrednosti tega atributa za vse entitete (povezave) tega tipa. Stolpcu relacije pravimo tudi polje (angl. field), vrstici pa podatkovni zapis (angl. record) (Kovačič et al., 2004, str. 129-137).

Glavni ključ relacije

Relacija ima lahko več kandidatov za ključ, od katerih izberemo enega za glavni ključ, ostali pa so nadomestni ključi. Glavni ključ ne more imeti v dveh vrsticah (dva podatkovna zapisa) iste vrednosti, označimo pa ga tako, da ga podčrtamo. Ostalim atributom pravimo neključni atributi, ki so odvisni od glavnega ključa. Če v relaciji nimamo atributa, ki bi sam zadoščal

zahtevam za glavni ključ, izberemo za glavni ključ ustrezno kombinacijo atributov, ki ji pravimo sestavljen glavni ključ.

Relacijska shema

Pri prikazu relacij ponavadi ne prikazujemo celotne tabele z vsemi podatki, saj nas pogosto zanima le struktura relacije (posebej pri načrtovanju), poleg tega pa se podatki, ki odražajo realnost, pogosto spreminjajo, pri čemer pa struktura relacije ostaja ista. Struktura relacije je dobro opredeljena s svojim imenom in seznamom atributov, kar prikažemo tako, da za imenom relacije v okroglem oklepaju naštejemo attribute relacije, glavni ključ pa podčrtamo. Tak zapis imenujemo relacijska shema, relacijska shema podatkovne baze pa bo praviloma vsebovala več takih shem za posamezne relacije. Relacijska shema podatkovne baze je hkrati njen načrt (z upoštevanjem, da je potrebno izdelati še t.i. fizični načrt, v katerem natančneje opredelimo fizične značilnosti, na primer tipe in velikosti posameznih atributov), saj opredeljuje, kako naj strukturiramo podatke: tabele, njihove attribute, glavne ključe itd.

Povezave

Podatki iz dveh ali več relacij so lahko med seboj povezani, v relacijskem modelu pa take podatke vsebinsko povezujemo s skupnimi atributi. Gre za tisto nujno podvajanje, ki ga v relacijskih podatkovnih bazah dovoljemo.

Povezave med entitetami v relacijskem modelu lahko prikažemo na dva načina:

1. Kadar med entitetnima tipoma obstaja povezava 1:N, potem lahko povezavo med ustreznima relacijama vzpostavimo tako, da v relacijo na strani N dodamo glavni ključ relacije na strani 1. Kadar imamo povezavo 1:1, pa je izbira, kam damo atribut, načeloma poljubna, pa vendar je v večini primerov ena od izbir boljša od druge.
2. Kadar med entitetnima tipoma obstaja povezava M:N, potem tvorimo novo relacijo, ki predstavlja prav povezavo med tema entitetnima tipoma. Glavni ključ te relacije bo sestavljen iz ključev relacij, ki ju povezuje.

Tuji ključ je neključni atribut ene relacije, ki je v drugi relaciji glavni ključ, označimo ga tako, da ga podčrtamo s prekinjeno črto. Povezave med entitetami lahko torej predstavimo s tujimi ključi ali z relacijo, ki ima sestavljen glavni ključ.

Značilnosti relacijskega podatkovnega modela (Grad, Kovačič, Barle, 1990, str. 54):

1. relacijski model je preprost za razumevanje,
2. temelji na čvrsti matematični podlagi, to je teoriji relacij, in teoriji normalizacije, s pomočjo katere odpravljamo razna protislovja in težave pri samem vzdrževanju podatkovne baze,
3. neodvisnost oblikovanja - dejanska fizična predstavitev je uporabniku skrita; uporabnik podatke vidi kot da bi bili shranjeni v dvodimenzionalnih tabelah, čeprav je lahko dejanska predstavitev zaradi učinkovitosti bistveno bolj zapletena,

4. poizvedovanje je praviloma počasnejše kot pri uporabi ostalih modelov, saj so povezave med podatki implicitne (skupni atributi), pri drugih modelih pa eksplicitne,
5. za poizvedovanje se uporablja standardni poizvedovalni jezik SQL (angl. Structured Query Language), kar omogoča enostaven prehod na uporabo drugega sistema za upravljanje podatkovnih baz.

Sistemi za upravljanje relacijskih podatkovnih baz

Relacijske podatkovne baze so se zaradi enostavnosti izredno razširile v praksi, tako da lahko danes izbiramo med večjim številom relacijskih SUPB, med katerimi velja omeniti Oracle, MS SQL, DB2, MySQL in PostgreSQL. Uporaba omenjenih sistemov temelji na arhitekturi odjemalec/strežnik, kjer je SUPB podatkovni strežnik, programi, ki dostopajo do podatkov, pa so odjemalci.

4.2.2.3. Večdimenzionalni model podatkov

Sheme relacijskih podatkovnih baz so že za relativno »majhna« problemska področja presenetljivo kompleksne, ta kompleksnost pa je za uporabnike na operativni ravni skrita za programi, s katerimi delajo, kar pomeni, da ti uporabniki sheme sploh ne poznajo oziroma jim je ni treba poznati. Na področju podpore odločanju, kjer je delo s podatkovnimi viri precej bolj nerutinsko (ad hoc poizvedbe), pa je stanje popolnoma drugačno, saj so poizvedbe po podatkovnih virih zelo raznolike, ni vnaprej pripravljenih programov, poznavanje sheme podatkovnega vira pa zato postane nujno. Drugi pomemben razlog za drugačno organizacijo podatkov pa je tudi sama počasnost poizvedovanja, ki pride še posebej do izraza pri delu odločevalcev, ki naenkrat delajo z ogromnimi količinami podatkov. Kot posledica teh slabosti relacijskega modela se v podatkovnih virih, ki so namenjeni podpori odločanju, uporablja večdimenzionalni podatkovni model (Kovačič et al., 2004, str. 138).

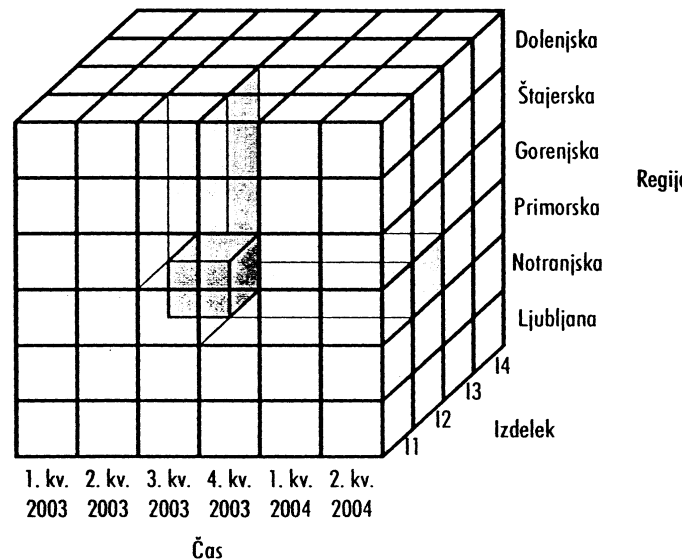
Večdimenzionalni model v primerjavi z relacijskim modelom omogoča hitrejši dostop do predhodno agregiranih, večdimenzionalnih podatkov. Zmanjšuje število fizičnih združevanj potrebnih za poizvedbe, uporaben je predvsem pri skladiščih podatkov, saj sloni na hitrem procesiranju zgodovinskih podatkov. Podvajanje podatkov se dopušča, še posebej shranjene seštete vrednosti, oblika diagrama pa je običajno zvezdasta (Špehar, 2004, str. 18).

Njegova bistvena značilnost je, da so sheme, ki jih lahko realiziramo, preproste in blizu načinu razmišljanja analitika/odločevalca. Dopušča organizacijo podatkov v obliki večdimenzionalne kocke. Iz Slike 5 (na strani 22), kjer je primer tridimenzionalne kocke, je razviden podatek o vrednosti prodaje izdelka I2 na Primorskem v tretjem kvartalu leta 2003.

Stranice kocke imenujemo tudi dimenzije, vsaka točka oziroma element kocke je na preseku koordinat, ki jih definirajo stranice kocke. Elementom kocke pravimo tudi mere (npr. vrednost prodaje za neko kombinacijo dimenzij). Posebej velja opozoriti na dimenzijo čas, ki je

značilna za podatkovne baze, namenjene podpori odločanju, redkeje pa jo srečamo v podatkovnih bazah, namenjenih podpori izvajanju operativnih poslovnih procesov, kjer nas zanimajo predvsem trenutne vrednosti podatkov.

Slika 5: Primer tridimenzionalne kocke



Vir: Kovačič et al., 2004, str. 139.

Za realizacijo večdimenzionalne sheme lahko uporabimo sisteme za upravljanje večdimenzionalnih podatkovnih baz, zelo pogosto pa večdimenzionalno shemo realiziramo kar z relacijsko podatkovno bazo. Vsaki dimenziji pripada ena tabela, ki jo imenujemo dimenzijska tabela (angl. dimension table). Tabela, v kateri shranjujemo podatke za vrednosti mer pri posamezni kombinaciji vrednosti dimenzijskih atributov, pa imenujemo tabela dejstev (angl. fact table). Tabela dejstev povezuje dimenzijske tabele, zato je njen ključ sestavljen iz glavnih ključev dimenzijskih tabel, praviloma pa ta vsebuje več mer. Običajno je zelo velika in se hitro spreminja, v nasprotju z njo pa so dimenzijske tabele majhne in se spreminjajo počasi (Kovačič et al., 2004, str. 140).

4.2.2.4. Objektni model podatkov

Razvoj svetovnega omrežja zahteva vključevanje nestrukturiranih podatkov kot so slike, video, zvok in grafični podatki, kar pomeni da morajo tudi moderne aplikacije omogočiti možnost definiranja poljubnih podatkovnih tipov. Vedno hujša konkurenca zahteva prilagodljivost podjetij spremembam v okolju, kar ima za posledico, da so tudi sami poslovni procesi in informacijski sistemi vedno bolj kompleksni in stalno v procesu reinženiranja, vse to pa vodi k pogostim spremembam tudi v informacijski podpori.

Pri iterativnem pristopu gradnje informacijskih sistemov, kjer se ta gradi po korakih, so objekti kot enotnost podatkov in operacij nad njimi osnova za takšen pristop. V prvi iteraciji lahko izdelamo le osnovne objekte, ki vključujejo le osnovne operacije nad atributi objekta, v naslednji iteraciji pa dodamo nove objekte, gradimo hierarhijo objektov, dodajamo operacije in tako postopoma gradimo kompleksen informacijski sistem. Podjetja se zavedajo prednosti takšnega pristopa in želijo imeti podporo takšnemu razvoju tudi s strani podatkovne baze (Komel, 1998, str. 21).

Tradicionalne metode razvoja ločujejo podatkovni in procesni vidik modeliranega realnega sveta, kar vodi do semantičnega neskladja med končnim proizvodom in procesom, katerega smo modelirali. Tako stanje je težko sprejemljivo in to je eden od pomembnejših razlogov, da je prišlo do razvoja novih metodologij razvoja programske opreme. V trenutku, ko v celoti uporabljamo le objektne metode, niso potrebna preslikovanja med različnimi fazami razvoja (analiza, načrtovanje, implementacija in modeliranje podatkovne baze), ampak razvijalci uporabljajo en sam, konsistenten konceptualni model. Uporaba enotnega konceptualnega modela vzpodbuja uporabo osrednje shrambe podatkov, kamor v fazi razvoja shranjujemo vse pridobljene informacije in jih uporabimo kadarkoli kasneje. Trend uporabe osrednjih shramb podatkov je še posebej dobro viden na področju orodij CASE, saj ta orodja že nekaj časa izkoriščajo osrednje shrambe podatkov kot osnovo za učinkovito delovanje celotnega procesa razvoja (Domajnko, Heričko, Rozman, 1997, str. 185-188).

Nekaj najpomembnejših pojmov na področju objektnih podatkovnih baz (Domajnko, Heričko, Rozman, 1997, str. 184):

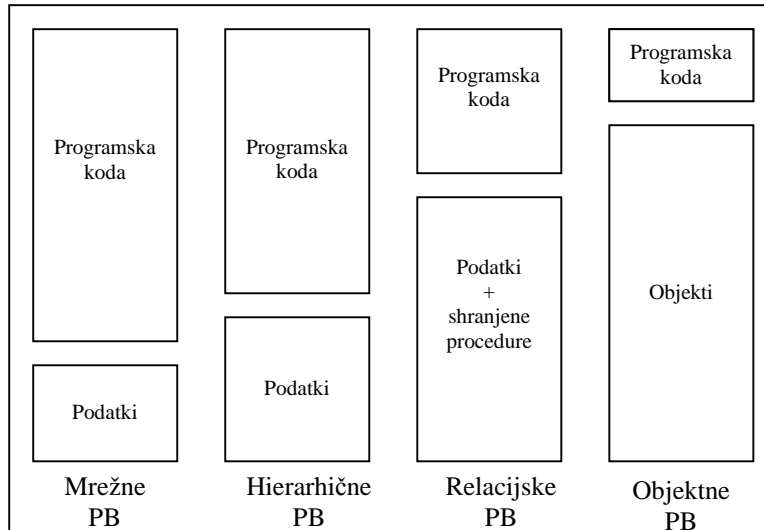
- *Objekt*: skupek podatkov in programske kode, ki zagotavlja konsistenco uporabe podatkov objekta. Je združitev podatkov in operacij oziroma metod nad podatki. Primer poslovnega objekta je naročilo, proizvod, proizvodni proces, zaposleni in podobno. Izvedba objekta je skrita pred njegovimi uporabniki, ki lahko pristopajo do podatkov le preko njegovih operacij.
- *Objektni identifikator* (angl. Object Identifier, OID): logično prirejena unikatna vrednost objektu v podatkovni bazi, ki zagotavlja nespremenjeno identiteto. Uporablja se za zagotavljanje konsistentnosti in omogoča enostavno sklicevanje objektov v podatkovni bazi.
- *Objektna referenca*: izrablja lastnost identifikacije s pomočjo OID in omogoča neposreden dostop do posameznih objektov.
- *Navigacija*: ena največjih prednosti, saj izkorišča poznavanje semantike podatkov z namenom hitrega in učinkovitega dostopa. Osnova navigacije je identiteta objektov, ki zagotavlja konsistenten dostop do podatkov, hkrati pa predstavlja temelj logičnega dostopa do objektov.
- *Kopičenje podatkov* (angl. Data Clustering): shranjevanje podatkov objekta ali povezanih objektov na fizično majhnem prostoru in pomnilnem mediju, kar skupaj z načelom lokalnosti omogoča hitrejši dostop do podatkov.

- *Dedovanje*: objekt določimo kot podobjekt obsoječega in ta deduje vse njegove lastnosti, dodane pa so mu še lastnosti značilne zanj.
- *Z ograjevanjem* ščitimo podatke in do njih pristopamo le preko operacij nad podatki, ki skupaj sestavljajo objekt.
- *Objektni podatkovni model*: semantiko vsebujoč podatkovni model, ki poleg vsebovanja podatkov omogoča tudi modeliranje procesov za uporabo teh podatkov.

Ena glavnih prednosti objektnih baz je možnost preslikave objekta iz realnega sveta v računalniško okolje brez potrebe po transformaciji v aplikativni in podatkovni model ter v nadaljno razgradnjo podatkovnega dela na elemente, ki so primerni za shranjevanje v obliki tabel relacijskih baz. Podatke različnih tipov objektne baze shranjujejo na »naraven« način, objektov torej ne razstavljajo do nivoja elementarnih podatkov. Med seboj so objekti v objektni podatkovni bazi povezani s pomočjo kazalcev, ki so podobni kazalcem mrežnih podatkovnih baz. S kazalci se doseže odlične zmogljivosti pri poizvedovanjih, saj so povezave zapisane v atributih objekta, kar ima za posledico mnogo manj vhodno/izhodnih operacij na disku in prometa po mreži. Tudi pisanje samih ad hoc poizvedb za končne uporabnike je enostavnejše, ker ni potrebno pisati zahtevnih pogojev povezovanja tabel, za kar je potrebno dobro poznavanje strukture baze (Komel, 1998, str. 22).

Ena izmed prednosti uporabe objektnih metodologij za razvoj sistemov in objektnih podatkovnih baz za zagotavljanje trajnosti je tudi zmanjšanje obsega programske kode, ki implementira posamezne vidike obdelave podatkov, saj z uporabo objektne podatkovne baze ni več potrebe po pretvorbi podatkov v obliko, ki bi jo tradicionalna podatkovna baza bila sposobna shraniti. Uporabljene objekte v eni aplikaciji lahko uporabimo v neki drugi, objekta torej ni potrebno kreirati od začetka, ga testirati in preverjati v obstoječem okolju. (Domajnko, Heričko, Rozman, 1997, str. 186). Na Sliki 6 je prikazana primerjava deleža kode v aplikacijah pri uporabi posameznih generacij podatkovnih baz.

Slika 6: Delež kode v aplikacijah pri uporabi posameznih generacij podatkovnih baz



Vir: Domajnko, Heričko, Rozman, 1997, str. 186.

Sistem za upravljanje objektnih podatkovnih baz je hkrati sistem objektov in sistem za upravljanje s podatkovnimi bazami. V prvi vrsti je slednji, kar pomeni, da zagotavlja trajno shranjevanje podatkov, preprečuje izgubo podatkov, omogoča hkratno delo večih uporabnikov brez bojazni za nekonsistentnost podatkov, učinkovito upravlja z velikimi količinami podatkov, omogoča administracijo podatkovne baze, po možnosti pa omogoča tudi distribucijo podatkov. Je tudi sistem objektov, ki podpira prej naštetosti in vključuje še združevanje kode in podatkov, skrivanje notranjih struktur, generiranje novih objektov in vzdržuje identiteto objektov (Domajnko, Heričko, Rozman, 1997, str. 185).

Objektni SQL je jezik objektnih podatkovnih baz, ki vključuje izrazno orientiran postopkovni jezik z visokonivojskim, deklarativnim jezikom povpraševanja, katerega je razmeroma enostavno optimizirati. Jezik OSQL združuje objektno orientirane lastnosti, katere najdemo v jezikih kot sta C++ in Smalltalk ter sposobnost povpraševanja, ki je nadgradnja SQL relacijskega jezika (Domajnko, Heričko, Rozman, 1997, str. 187).

Področja, kjer objektno podatkovne baze izkoriščajo naravno prednost pred tradicionalnimi so področja tehničnih aplikacij (CAD/CAM in CAE), kjer gre za shranjevanje ogromnega števila razmeroma majhnih, medsebojno močno povezanih objektov, multimedijske aplikacije, kjer shranjujemo razmeroma majhno število velikih objektov in področje Interneta, kjer navzkrižna sklicevanja dinamično kreiranih strani izkoriščajo veliko prednost objektnih podatkovnih baz.

3.2.2.5. Objektno-relacijski model podatkov

Večina današnjih poslovnih informacijskih sistemov temelji na relacijskih bazah podatkov in obdelave tečejo nad strukturiranimi podatki. Prehod oziroma razvoj podatkovnih baz v smeri

objektnih baz pa je pogojen z zmožnostjo objektnih baz, da zagotovijo vse lastnosti, ki jih nudijo današnje relacijske. Še več, velika večina uporabnikov želi predvsem nadgrajevati sedanje stanje in niso pripravljeni zaradi tehnološkega preskoka graditi informacijskega sistema znova. To postavlja jasne zahteve vsem ponudnikom pri razvoju podatkovnih baz:

- zagotoviti obstoječo zmožljivost, zanesljivost in hitrost, ki jo nudijo relacijske baze,
- zagotoviti možnost vključevanja objektov in povezovanje novih rešitev z obstoječimi.

Podatkovne baze, ki izpolnjujejo zgoraj naveden zahteve, so po arhitekturi kombinacija objektnih in relacijskih in jih zato imenujemo objektno-relacijske podatkovne baze (Komel, 1998, str. 20).

V relacijskem okolju lahko vsak trenutek spreminjamo način povezave med tabelami in gradimo različne izpise. Povezovanje objektov med seboj na drugačen način, kot je določeno pri graditvi objektov, pomeni precej zahtevno nalogo, zato je pri določanju povezav med objekti pri kreiranju baze potrebno izredno dobro poznavanje potrebe aplikacij, ki se bodo izvajale nad tako povezanimi objekti. Ker pa narava poslovnih aplikacij zahteva fleksibilnost poizvedovanja po različnih informacijah, fleksibilnost relacijske baze v poslovnih okoljih nedvomno odtehta prednosti odzivnih časov objektnih baz, saj objektno baze ne dosegajo zmožljivosti prevladujočih relacijskih baz na področju obdelave strukturiranih podatkov.

Prilagodljivost podatkovne baze velikosti podatkov in številu hkratnih uporabnikov je eno pomembnih področij, kjer objektno baze še niso prava alternativa relacijskim. Tudi sama infrastruktura relacijskih baz v smislu varovanja podatkov v primeru programskih ali strojnih napak, lastnost varovanja pred nepooblaščenim pristopom do podatkov in upravljanja s podatkovno bazo trenutno močno prekaša objektno baze. Poleg naštetega so današnji uporabniki relacijske tehnologije investirali ogromna sredstva v programsko opremo in ljudi, zato so večinoma pripravljeni sprejeti samo takšne rešitve, ki gradijo razvoj na osnovi obstoječega (Komel, 1998, str. 24).

Zaradi naštetih pomanjkljivosti ene in druge strukture podatkov v podatkovnih bazah je prišlo do združevanja relacijskih in objektnih podatkovnih baz. Objektno-relacijska baza ima vse lastnosti relacijske baze, dodane pa so možnosti definiranja bogatih podatkovnih tipov, ki omogočajo shranjevanje objektov v isti bazi s skalarnimi tipi podatkov. Omogoča nam torej upravljanje s standardnimi večpredstavnimi podatki, kot so slike, zvok, geografski podatki in podobno.

4.2.3. TEHNOLOŠKO NEODVISNI MODELI PODATKOV

Z dosedaj opisanimi podatki lahko učinkovito predstavimo organizacijo podatkov, predvsem na ravni notranjega modela podatkov. Vendar je v svetu prevladalo mnenje, da je za predstavitev zunanjega in konceptualnega modela podatkov primernejša oblika modela

podatkov, ki je neodvisna od posamezne notranje predstavitve podatkov, ki je pod vplivom uporabljenega sistema za upravljanje podatkovnih baz. Enovita oblika modela podatkov za razvoj zunanjega in konceptualnega modela podatkov je bila razvita z namenom opremiti znane modele podatkov z dodatnimi informacijami, posebno s tistimi, ki ustrezajo ravni logičnega modela podatkov, ter razviti programska orodja, ki so neodvisna od uporabljene tehnologije pri konceptualni zasnovi. Najpogosteje uporabljen model, ki je bil razvit v ta namen, je model entitet-povezav oziroma ER model.

4.2.3.1. Model entitet-povezav

Model entitet povezav (angl. Entity Relationship Model, ERM) je ena izmed v praksi najpogosteje uporabljenih tehnik, s katerimi grafično prikažemo pogled na obravnavani del realnega sveta. Je relativno enostaven za razumevanje, hkrati pa omogoča dovolj široko paleto možnosti formalnega (nedvoumnega) izražanja strukture podatkov in poslovnih pravil. Posebej pomemben je pri komunikaciji med analitiki, izvajalci poslovnih procesov in uporabniki podatkov za poslovno odločanje. Model je »poslovno« usmerjen, kar pomeni, da vsebino in strukturo sistema nadzoruje uporabnik in ne informatik. Poudarja se torej uporabnikove zahteve in ne rešitve in omejitve, ki jih postavlja tehnologija. Uporablja se poslovna in ne tehnična terminologija (Kovačič et al., 2004, str. 101-113).

Entiteta

Je predmet ali dogodek, ki ga obravnavamo. Nekateri avtorji jo definirajo kot pojav entitetnega tipa (Brathwaite, 1991, str. 15). Kot primer lahko vzamemo entitetni tip DELAVEC, kjer entiteto predstavlja NOVAK JANEZ.

Atribut

Je neka lastnost entitete, pri čemer upoštevamo le attribute, ki so predmet našega zanimanja, torej upoštevamo kontekst.

Vrednost atributa

Vsak atribut ima v določenem trenutku nobeno, eno ali več vrednosti. Kadar jih ima lahko več, govorimo o večvrednostnem atributu, v primeru, ko v določenem trenutku neki atribut nima vrednosti (ali je ne poznamo), to označimo z vrednostjo NULL, ki pa je ne smemo zamenjevati z vrednostjo 0 (številski podatki) ali vrednostjo prazen niz (tekstovni podatki), saj v teh primerih ne gre za manjkajoče vrednosti.

Domena

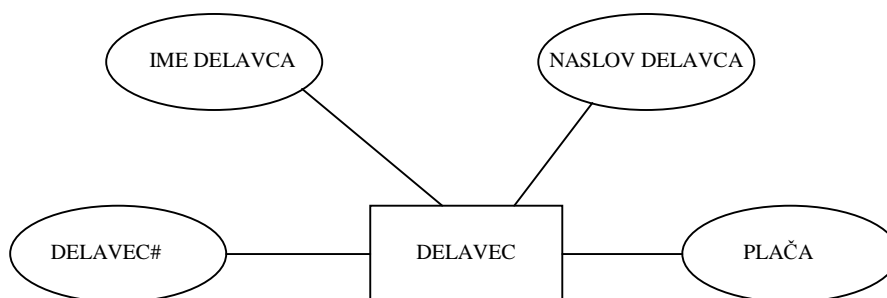
Je množica dopustnih vrednosti nekega atributa. Definiranje domen je pomembno pri nadzoru nad nenamernimi napakami pri vnosu podatkov, saj sodobna informacijska tehnologija omogoča avtomatičen nadzor nad tem, ali je vrednost atributa, ki jo je vnesel uporabnik, izven domene ali ne.

Sestavljen atribut je tak, ki ga lahko naprej razdelimo na manjše enote. Nasprotno pa atributu, ki ga ni smiselno razbijati na manjše enote, pravimo **elementarni atribut**.

Izpeljani atribut pa je tisti, katerega vrednosti lahko izpeljemo (izračunamo) iz ostalih atributov. Ker je to mogoče v vsakem trenutku, jih ni potrebno shranjevati neposredno, saj se držimo načela, da želimo vsako dejstvo shraniti na en sam način.

Entitetni tip predstavlja vzorec za entitete z enakimi (ali vsaj podobnimi) atributi. Pri načrtovanju podatkov namreč večinoma ne obravnavamo vsake entitete posebej, saj bi to pomenilo preveliko kompleksnost, hkrati pa to tudi ni potrebno, pač pa preučujemo entitetne tipe in povezave med njimi.

Slika 7: Grafični prikaz entitetnega tipa z atributi



Vir: Kovačič et al., 2004, str. 106.

Entitetni tip grafično prikažemo s pravokotnikom, v katerega vpišemo ime entitetnega tipa, za grafični prikaz atributov entitetnega tipa pa uporabljamo elipse, v katerih je zapisano ime atributa. Grafični prikaz entitetnega tipa z atributi prikazuje Slika 7.

Kandidat za ključ

Atribut (ali več skupaj), za katerega ne moreta imeti v nobenem trenutku nobeni dve entiteti istega entitetnega tipa enake vrednosti, imenujemo kandidat za ključ. Kandidat za ključ torej enolično identificira entiteto znotraj entitetnega tipa, ker je namenjen identifikaciji pa ne sme imeti vrednosti NULL.

Ključ

Kadar imamo več kandidatov za ključ, med njimi izberemo enega, ki ga v nadaljevanju uporabljamo za to, da med seboj ločimo entitete znotraj entitetnega tipa. Izbrani kandidat

imenujemo glavni ključ (angl. Primary Key). Na odločitev pri izbiri glavnega ključa lahko vplivajo različni parametri; prednost imajo enostavnejši kandidati (krajše vrednosti in kandidati, ki niso sestavljeni iz več atributov), izogibamo se uporabi »govorečih« šifer (šifre, iz katerih se da razbrati podatke osebne ali kako drugače občutljive narave), upoštevati je potrebno varnostne vidike itd. Še posebej moramo biti pozorni, da se vrednost glavnega ključa v življenjski dobi entitete ne spreminja, saj bi spremenjena vrednost atributa, ki identificira entiteto, pomenila tudi drugo entiteto. V ER diagramu ime ključnega atributa podčrtamo.

Sestavljeni ključ

Nekaterih entitet ni mogoče enolično opredeliti z enim samim atributom oz. z glavnim ključem, ki je en sam atribut. Takrat uporabimo glavni ključ, ki je sestavljen iz dveh ali več atributov. Takšen ključ imenujemo sestavljeni ključ (angl. Concatenated Key). V ER diagramu sestavljeni ključ prikažemo tako, da podčrtamo imena vseh atributov, ki sestavljajo glavni ključ ali pa tako, da v eni elipsi zapišemo podčrtana imena vseh sestavnih delov ključa.

Povezava in tip povezave

Za analizo problemskega področja z vidika podatkov je pomembno tudi ugotavljanje povezav med entitetami. Vendar pa tako kot v primeru entitet pri načrtovanju ne modeliramo posamezne entitete, tudi tu ne analiziramo povezav med posameznimi entitetami, temveč povezave med entitetnimi tipi. Tudi tu govorimo o tipu povezave med entitetami dveh ali več entitetnih tipov. V grafični obliki tipe povezav prikazujemo s črtami med pravokotniki, ki prikazujejo entitetne tipe.

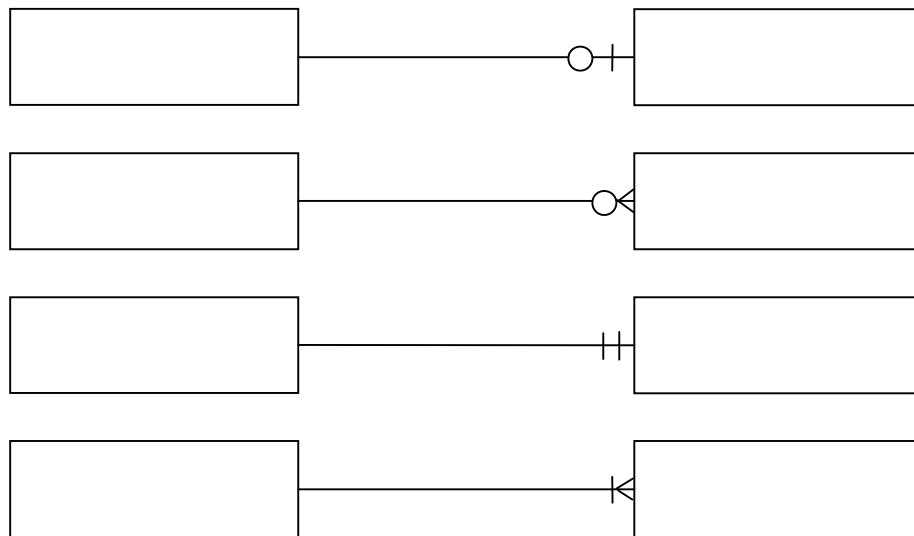
Kardinalnost povezav

Poznavanje in razumevanje povezav med entitetnimi tipi je potrebno načrtovalcem informacijskih sistemov, ker pa se v povezavah odražajo poslovna pravila, jih lahko analizira le s pomočjo poznavalca problemskega področja.

Pri vsaki povezavi se sprašujemo po spodnji in zgornji meji številčnosti (kardinalnosti) povezave. Če opazujemo povezavo med entitetnim tipom A in entitetnim tipom B, potem imamo za povezavo v smeri iz A v B štiri mogoče kombinacije:

- entiteta iz A ni nujno povezana z neko entiteto iz B (spodnja meja je 0), če pa je, je z največ eno entiteto iz B (zgornja meja je 1),
- entiteta iz A ni nujno povezana z neko entiteto iz B (spodnja meja je 0), lahko pa je tudi z več entitetami iz B (zgornja meja je mnogo, N),
- entiteta iz A mora biti v vsakem trenutku povezana z vsaj eno in največ eno entiteto iz B (spodnja in zgornja meja je 1),
- entiteta iz A mora biti v vsakem trenutku povezana z vsaj eno entiteto iz B (spodnja meja je 1), lahko pa je tudi z več entitetami iz B (zgornja meja je N).

Slika 8: Grafični prikaz kombinacij



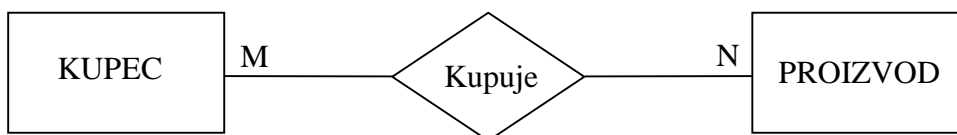
Vir: Kovačič et al., 2004, str. 110.

Grafično spodnje in zgornje meje povezav označimo na povezavi s simboli (Slika 8 na strani 29):

- krožec za 0,
- črtica prek povezave za 1,
- tri razprte črtice za N.

Neredko v praksi srečamo tudi uporabo drugačnih grafičnih oznak v modelih ER. Tako včasih povezave dodatno označimo z likom romba, kardinalnost označujemo tako, da na črte, ki simbolizirajo povezave, zapišemo oznake kar z znaki 0, 1 ali N (Slika 9), spodnje meje kardinalnosti označimo tako, da povezavo narišemo s polno (obvezna povezava) oziroma črtkano povezavo (neobvezna povezava) itd. Zelo pogosto attribute, namesto elipse okoli entitetnega tipa, zapišemo kar v pravokotnik, ki simbolizira entitetni tip, pri čemer ime entitetnega tipa ločimo od atributov s črto.

Slika 9: Primer drugačnega grafičnega prikaza ER diagrama



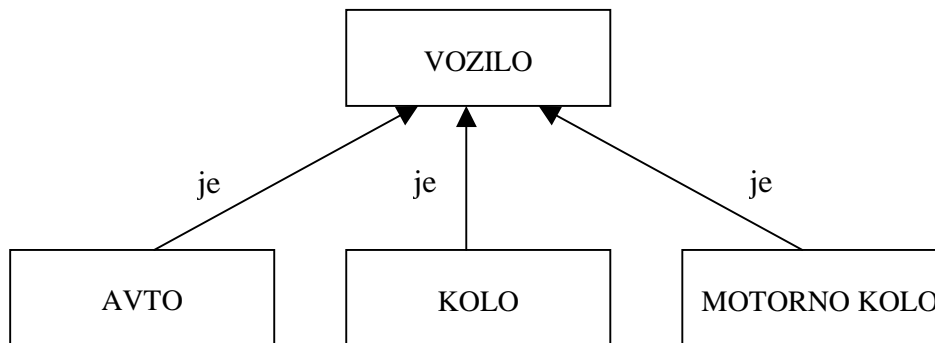
Vir: Lucas, 1992, str. 219.

Generalizacija

V praksi se pogosto posebej obravnava nekatere tipe povezav, še posebej pa je zanimiva povezava, ki označuje generalizacijo. Obstajajo primeri entitetnih tipov, ki so bolj splošni od

drugih, zajemajo entitete iz več različnih entitetnih tipov in tako tvorijo entitetni tip, ki je »obsežnejši« oziroma bolj splošen.

Slika 10: Grafični prikaz generalizacije



Vir: Vintar, 1990, str. 42.

Generalizacija je koncept abstrakcije, pri katerem se posameznim entitetnim tipom priredi splošnejši tip na višjem nivoju, kot prikazuje Slika 10 na strani 30 (Vintar, 1990, str. 42). Vsaka entiteta, ki je tipa AVTO, je tudi entiteta tipa VOZILO. Enako velja, da je vsaka entiteta, ki je tipa KOLO in MOTORNO KOLO, hkrati tudi tipa VOZILO. Entitetni tip VOZILO je torej bolj splošen od tipov AVTO, KOLO in MOTORNO KOLO.

4.2.2.2. Semantični model

Cilj modeliranja na zunanji in konceptualni ravni je, da popišemo, spoznamo in razumemo podatke neke organizacije. Poudarek je tako na pomenu in ne na strukturi ter s tem ugotoviti zahteve baze podatkov. Model podatkov, ki naj zadosti temu namenu oziroma, ki je usmerjen na dokumentiranje pomena podatkov, pa mora seveda vsebovati tudi dovolj informacij za potrebe kasnejšega strukturiranja podatkov. Nekatero informacije, ki so potrebne za opredelitev pomena posameznih pojmov, kot so objekti – entitete in povezave med njimi, so naslednje (Grad, Kovačič, Barle, 1996, str. 77):

1. *Pravila glede pripadnosti povezavam med objekti – entitetami:* vsak objekt mora obstajati neodvisno od drugih objektov; obstoj enega objekta je pogojen z obstojem drugega (ti dve pravili sta upoštevani v ER modelu); povezava je pogojena z nekim kriterijem (samo naročila kupcev so povezana s kupci, ne pa tudi naročila trgovin in skladišč); nek objekt je trajno povezan z drugim objektom in ta povezava se ne more menjati (neko naročilo se nanaša le na določenega kupca – vse dokler ni izničeno); povezava objektov je le začasna (pripadnost delavca določenemu oddelku); in končno, povezava objektov je selektivna (skladišče vsebuje proizvode ali surovine, vendar ne obojih hkrati).

2. *Povzetki splošnosti objektov* (angl. *entity abstractions*): to so objekti, katerih primeri predstavljajo posplošenja ali vrste (tipe) drugih objektov (na primer, objekt DELOVNI NALOG vsebuje različne vrste NAROČILA, kot na primer naročilo za kupca in naročilo skladišča); vsak posamezen primer objekta DELOVNI NALOG lahko vsebuje svojstva-attribute, kot sta skupna-vrednost in število-aktivnih-naročil.
3. *Agregacije objektov* (angl. *entity aggregations*): to so objekti, katerih primeri predstavljajo zbir(ko) drugih objektov (na primer, objekt BLAGOVNA SKUPINA, katerega posamezen primer vsebuje množico-niz primerov objekta PROIZVOD: primer, ki vsebuje množico PROIZVODOV sedemdesetih let).
4. *Podmnožice objektov* (angl. *entity subsets*): to so objekti, ki predstavljajo podmnožice drugih objektov (na primer, ZAPRTA NAROČILA so podmnožica objekta NAROČILA; le-ta imajo svoje lastne dodatne podatkovne postavke datum-zaprtja in datum-izstavitve-računa).

Upoštevanje naštetih potrebnih elementov informacij, sta Hammer in McLeod (1978) razvila semantični model podatkov, ki podaja te in še druge opisne informacije o objektih. Tudi semantični model podatkov (SMP), podobno kot relacijski model, ni prikladen za predstavitev v obliki diagrama, temveč v obliki definicij. SMP omogoča predstavitev skoraj vseh značilnosti podatkov, ki si jih snovalec baze podatkov lahko zamisli. V načelu ga je možno prirediti tako, da vsebuje katerokoli željeno opisno določilo. Zaradi vseh teh možnih oblik posredovanja informacij pa lahko postane opis baze podatkov s pomočjo SMP zelo dolg. To je najpopolnejši model podatkov, ki je na voljo za definiranje baz podatkov na zunanji in konceptualni ravni.

5. PODATKOVNI MODEL PROIZVODNEGA PODJETJA

Z izdelavo globalnega podatkovnega modela bom poskusil podati osnovo za kasnejšo gradnjo podatkovne baze in njihovim potrebam prilagojenega informacijskega sistema. Upoštevati sem moral omejene vire podjetja, saj je na koncu koncev v podjetju redno zaposlenih le pet ljudi, tako da si zaenkrat ne morejo privoščiti celotne uprave, z računovodsko funkcijo, informacijsko službo itd. Veliko dokumentov sem moral zaradi morebitnih potreb po različnih vidikih na stroške in zaradi same sledljivosti materiala, polproizvodov in končnih proizvodov vpeljati na novo in s tem poskrbeti za celovitost zajemanja relevantnih podatkov. Da pa ne bi vse skupaj ostalo le na papirju v obliki modela, sem v nadaljevanju predstavil tudi okvirni prototip podatkovne baze z izdelavo same relacijske sheme, saj poleg dejstva, da imajo v podjetju vsi računalniki naložen programski paket Microsoft Office in z njim vred program Access, se dva izmed zaposlenih lahko pohvalita z dosedanjimi izkušnjami na področju upravljanja z relacijskimi podatkovnimi bazami, še posebej z Accessovimi podatkovnimi bazami. Za analizo poslovanja in s tem definiranjem informacijskih potreb sem si pomagal z intervjuji in pregledom obstoječih dokumentov podjetja.

5.1. KRATEK OPIS PODJETJA

Obravnavano podjetje je v lasti fizičnih oseb in je pričelo delovati leta 2002. Registrirano je kot družba za avtomatizacijo proizvodnih procesov, trgovino in proizvodnjo. Bolj natančno se ukvarja z avtomatizacijo proizvodnih procesov, razvojem elektronskih komponent in programske opreme ter proizvodnjo elektronskih komponent in merilnih sond. Podjetje je začelo delovati kot hčerinsko podjetje, kar pomeni, da je njegova glavna dejavnost bila in v bistvu še vedno je, izdelava varilnih mask z avtomatičnim preklapljanjem prepustnosti svetlobe. Matično podjetje mu preskrbi optične sklope, sami pa nato te sklope po opravljenem pregledu kakovosti opremijo z elektroniko, jih vgradijo v maske in po potrebi tudi tržijo, kar pa je prej izjema kot pravilo, saj dolgoročne pogodbe s strankami sklepa matično podjetje, obravnavano podjetje pa se tako podanim pogojem poslovanja uglavnem le prilagaja. V podjetju je redno zaposlenih pet ljudi, veliko pa si pomagajo s študenti.

Obe podjetji sta nastali v okviru Tehnološkega parka Ljubljana (TPL) v sodelovanju z Institutom Jožef Stefan (IJS), kar pomeni, da je IJS poskrbel za Raziskave in Razvoj (R&R), torej tehnologijo, država pa preko TPL z izjemo začetnega kapitala za vse ostalo, od subvencij in pomoči pri promociji izdelkov v tujini do garancij in poroštev pri različnih posojilih. Rezultat teh skupnih naporov v okviru projekta regionalnega podjetniškega inkubatorja je nastanek blagovne znamke, ki sodi v sam vrh svetovne ponudbe tovrstnih izdelkov. Kar 95 odstotkov svojih izdelkov izvozijo, pohvalijo pa se lahko s 60 odstotnim tržnim deležem v EU in kar 30 odstotnim svetovnim tržnim deležem.

Ker si v obravnavanem podjetju želijo zmanjšati odvisnost svojega poslovanja od matičnega podjetja, so s sodelovanjem IJS pričeli razvijati ter kasneje proizvajati in tržiti lastne izdelke pod lastno blagovno znamko. Razvili so na primer senzor za merjenje pretokov plinov in tekočin, ki je eden najbolj konkurenčnih merilnikov pretoka na tržišču Evrope in Amerike. Analiza trga jih je med drugim privedla tudi do izdelave natančnega sistema merjenja in doziranja pijač v gostinstvu in živilski industriji (proizvodnja alkoholnih in nealkoholnih pijač). Osnovni cilj podjetja je prodor na zahtevne evropske trge. Vsi izdelki, ki jih proizvajajo, so izdelani po veljavnih normativih in lahko nadomestijo podobne izdelke iz uvoza, vendar pa je njihova konkurenčna prednost pred uvoženimi izdelki nižja cena in prodaja izdelka skupaj s storitvijo vgradnje, vzdrževanja in izobraževanja uporabnika. Vsi proizvodi in izdelavni načrti so plod lastnega razvoja in domačega znanja. Osnovne sestavne dele izdelujejo kooperanti po njihovi izdelavni dokumentaciji, sami pa izvajajo sestavljanje, umerjanje, testiranje in končno kontrolo.

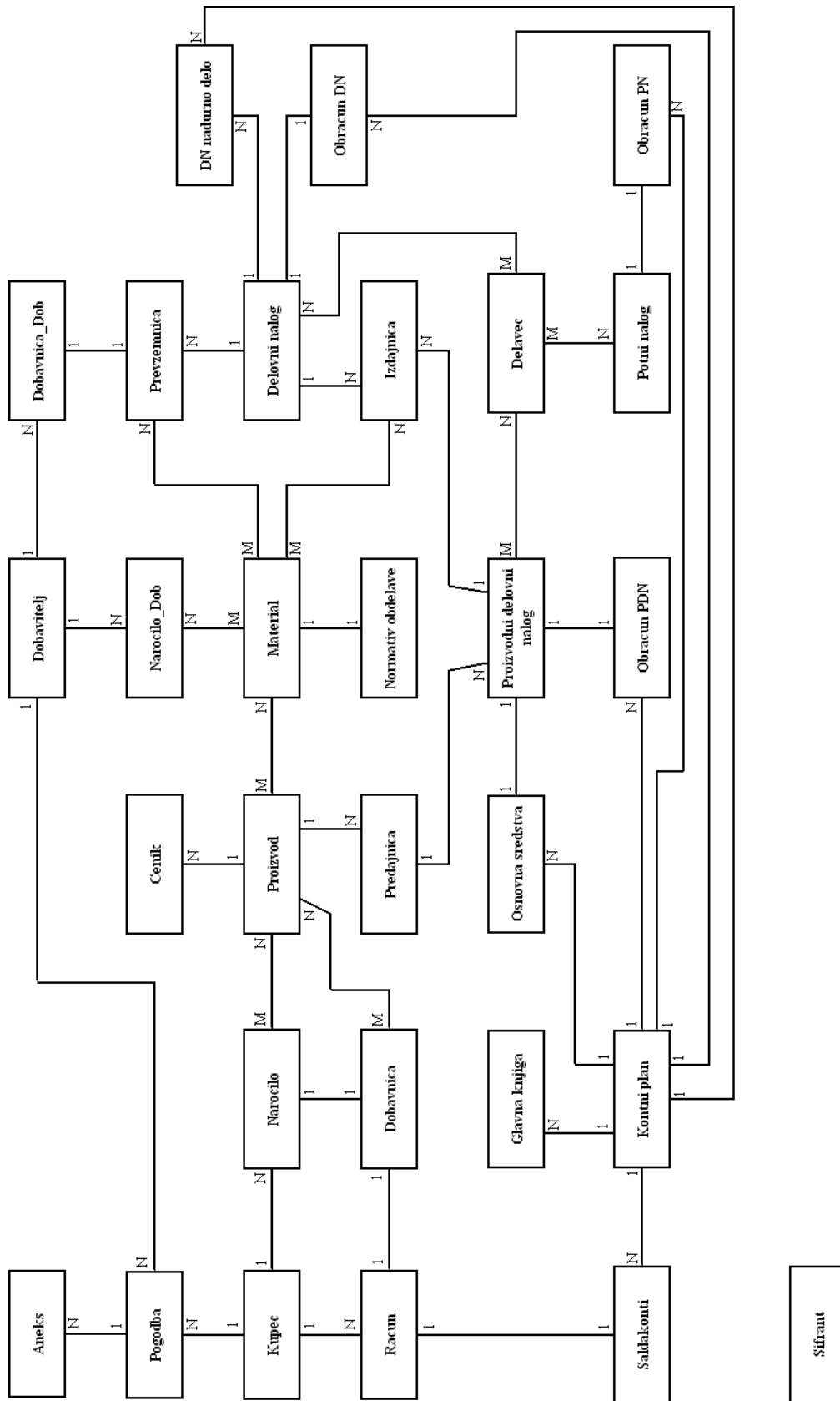
Obravnavano podjetje nima svojega informacijskega sistema, sicer zbirajo podatke in izdelujejo razna poročila, ki pa jih nato le posredujejo matičnem podjetju. V matičnem podjetju poskrbijo za računovodstvo in sam vnos podatkov v njihovo podatkovno bazo. Zaradi širjenja dejavnosti, novih izdelkov in planiranih bodočih novosti v samih postopkih

proizvodnje, pa se je izkazala potreba po lastnem informacijskem sistemu oziroma podatkovni bazi, ki bi upoštevala specifiko njim lastnega poslovanja. Prisiljeni so namreč prirežati podatke, da bi zadostili zahtevam za vnos v podatkovno bazo matičnega podjetja, s tem pa v neizogibno izgubljanje informacij in hranjenju relevantnih podatkov na vseh mogočih nosilcih, ki se običajno nahajajo v neposredni bližini delovnega mesta, kjer se ti potrebujejo. Posledica dejstva, da v obravnavanem podjetju nimajo dokumentov, s pomočjo katerih bi bilo omogočeno učinkovito spremljanje stanja zalog, je fizično štetje, ki se ga opravi pred vsakim večjim nakupom pri dobaviteljih. Ponavadi do naročila pri dobaviteljih pride celo šele takrat, ko delavci sami ugotovijo, da ne morejo nadaljevati dela zaradi nezadostnih zalog materiala.

5.2. GLOBALNI PODATKOVNI MODEL PODJETJA

Na sliki 11 (stran 34) je predstavljen globalni model podjetja, ki ga razlagam podrobneje v nadaljevanju.

Slika 11: Globalni podatkovni model obravnavanega podjetja



Vir: Lastna opazovanja.

Pri izdelavi globalnega podatkovnega modela sem si pomagal z izsledki analize, ki sem jo izvedel s pomočjo intervjujev in pregledom obstoječe dokumentacije. V podjetju sem bil tudi zaposlen kot študent. Intervjuje sem izvedel s tremi osebami, ki so mi pomagali dopolniti moje poznavanje podjetja, z dokumentacijo pa sem imel v mislih prejete račune, naročila, izdane dobavnice, potni nalogi in podobno.

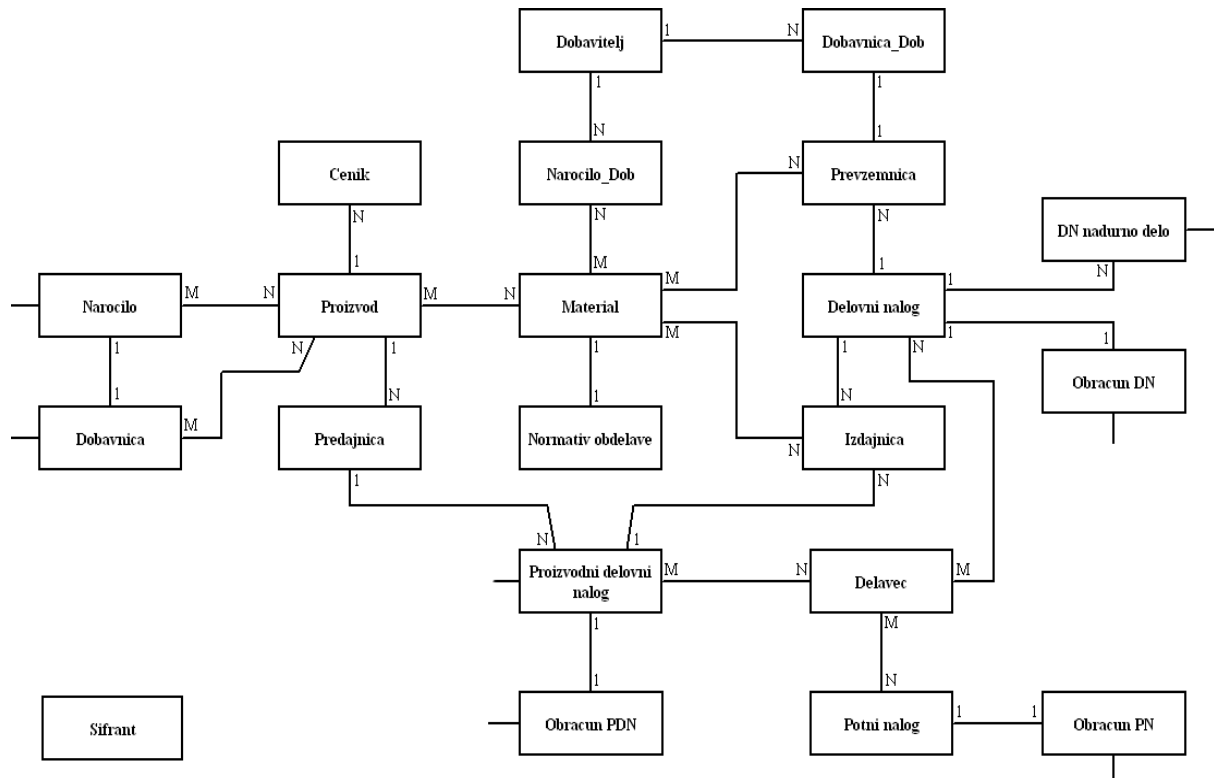
Vsako podjetje namreč prodaja svoje storitve ali izdelke, ima torej opravka s kupci, naročili, računi, pogodbami in tako naprej. Opredelitev podatkov, ki jih je potrebno zajemati v povezavi s temi entitetnimi tipi, je tako vitalnega pomena. Tudi v obravnavanem podjetju ni nič drugače, celo več, skoraj vsak izdelek je prirejen po meri kupca, od malih variant v samem izgledu, kot posledica uporabe drugačnih materialov, dimenzij, tiskanja itn, do bistveno drugačnih izdelkov, kot posledica uvedbe novih funkcionalnosti izdelka, ki širijo njegov spekter uporabe. S temi kupci so vzdrževane obnavljajoče se srednjeročne pogodbe o mesečnih naročilih, praktično vsak izmed njih pa ima, kot že povedano, svoje zahteve glede njim dobavljenega izdelka. To dejstvo pa je v podjetju razvidno že iz samega poimenovanja svojih polizdelkov, šifer materiala in podobno, ki je skladno z imenom kupca ali pa z nečim, ki je kako drugače z njim povezano (država, dimenzija, barva itd.). Entitetni tip Kupec je tako osrednjega pomena na strani 34 predstavljenem globalnem podatkovnem modelu, povezan z entitetnim tipom Pogodba, Racun in Narocilo. V vseh primerih gre za 1:N povezavo, saj se kupec lahko nahaja na več računih, naročilih in pogodbah, medtem ko je na enem računu, naročilu ali pogodbi lahko le en kupec.

Iz globalnega podatkovnega modela so razvidna tudi stičišča z računovodsko funkcijo, ki pa je za podjetje kljub dejstvu, da sami nimajo računovodij (torej zunanje izvajanje - outsourcing), še vedno pomembno že zaradi tega, da vedo kateri podatki in kako so priskrbljeni vsi podatki, ki jih računovodstvo matičnega podjetja potrebuje. Opredelitev atributov in vrednosti teh so v glavnem domena izbranih metod računovodskega obračunavanja (stroškov zalog, stroškov končnih proizvodov, stroškov dela, amortizacije itd). in same organizacije računovodstva (poimenovanja kontov, program, ki se za spremljanje računovodstva uporablja in podobno), ter jih kot takih ne obravnavam. Ti entitetni tipi so Saldakonti, Obracun DN, DN nadurno delo, Obracun PN, Obracun PDN, Osnovna sredstva, Kontni plan in Glavna knjiga.

Ker je poudarek mojega diplomskega dela na izdelavi podatkovnega modela proizvodnega podjetja, je na Sliki 12 (stran 36) predstavljen izsek globalnega podatkovnega modela, ki se dotika ravno problematike proizvodnje in zajemanja podatkov za podporo tega poslovnega procesa. Obravnavanem podjetju mora biti namreč omogočen vpogled v stanje zalog materiala in končnih proizvodov, različni pogledi na stroške zalog in stroške materiala (stroški prodanih proizvodov, stroški, ki so še vedno v procesu itn.), dodana vrednost posameznih aktivnosti in nadzor zaposlenih z vidika spremljanja njihove produktivnosti s primerjanjem njihovih rezultatov dela z normativi obdelave. Z namenom doseči ravnokar omenjene cilje, sem moral uvesti nekaj novih dokumentov, ki vsebujejo vse relevantne podatke, pri tem pa

upoštevati tudi specifično podjetja, ki je na eni strani hčerinsko podjetje, po drugi pa tudi samo sodeluje s kooperanti, ki jim nemalokrat proizvajajo in dobavljajo material in polproizvode, ki so jih praviloma zmožni proizvesti tudi sami.

Slika 12: Izsek globalnega podatkovnega modela



Vir: Lastna opazovanja.

Material, kot osrednji entitetni tip na sliki 12, je najprej povezan na levi strani z entitetnim tipom Proizvod. Tip povezave je M:N, saj je določen proizvod praviloma sestavljen iz različnih materialov, en in isti material pa se nahaja v več različnih proizvodih.

Na Material se navezuje tudi entitetni tip Normativ obdelave, s pomočjo katerega se zajemajo podatki o normah obdelave materiala, razne opombe, kot so npr. pogoji obdelave (v smislu delovnih prostorov), specifikacija hrambe, morebitne kosovnice itd.

M:N povezavo ima Material tudi z entitetnim tipom Narocilo_Dob, saj je na posamičnem naročilu podjetja dobaviteljem za dobavo materiala in polproizvodov lahko več postavk naročenega materiala in polproizvodov (odvisno od dobavitelja), prav tako pa se določen material oz. polproizvod lahko nahaja na več različnih naročilih. Entitetni tip Narocilo_Dob je nujen za spremljanje in upravljanje že naročenega materiala, ki pa lahko tudi še ni bil dobavljen. V tem primeru ne moremo količinsko povečati zaloga, je pa že naročen material potrebno upoštevati pri novih naročilih. Z analizo podatkov, ki se bodo zajemali s tem

entitetnim tipom, pa bodo v obravnavanem podjetju lahko tudi ocenjevali dobavne roke svojih dobaviteljev ter njihovo zanesljivost in s tem dobili neko osnovo za kriterij pri izbiri bodočih poslovnih partnerjev.

V podjetju so sami zmožni proizvesti večino polizdelkov, ki jih potrebujejo za izdelavo proizvoda, vendar pa ga za potrebe določenih kupcev tudi dobavljajo. Podlaga za evidentiranje količinskega povečanja stanja zalog materiala in polproizvodov je uvedba dokumenta Prezemnica, za katerega nastanek sta lahko dve osnovi. Prva v primeru dobavljenega materiala oziroma polproizvodov (povezava z entitetnim tipom Dobavnica_Dob), druga pa v primeru lastne proizvodnje oziroma obdelave (povezava z entitetnim tipom Delovni nalog). Uvedba dokumenta Izdajnica ima ravno nasprotno vlogo od prevzemnice, služi namreč evidentiranju količinskega zmanjšanja stanja zalog materiala in polproizvodov. Tako kot v primeru Prezemnica, sta tudi v primeru Izdajnica lahko dve osnovi za nastanek tega dokumenta. Prvi v primeru, ko se material vzame iz skladišča za namene obdelave oz. proizvodnje polizdelkov (povezava z entitetnim tipom Delovni nalog), drugi pa v primeru, ko se material in polproizvodi vzamejo iz skladišča z namenom dokončanja proizvodov in s tem vplivajo na zalogo končnih izdelkov ali neposredno na stroške prodanih izdelkov, torej odhodke (povezava z entitetnim tipom Proizvodni delovni nalog).

Razlika med entitetnima tipoma Delovni nalog in Proizvodni delovni nalog je v tem, da je prvi dokument povezan z materialom in polproizvodi, slednji pa s končnimi proizvodi in se nanaša na aktivnosti sestavljanja, umerjanja, testiranja in končne kontrole proizvodov, ki so tik pred dobavo naročnikom ali pa pripravljene na hrambo v skladišču končnih izdelkov. Ravno zaradi tega je smotna uvedba dokumenta imenovanega Predajnica, ki dejansko predstavlja količinsko povečanje stanja zalog končnih proizvodov in temelji na ustreznem proizvodnem delovnem nalogu. Da pa bi lahko spremljali stanje zalog končnih proizvodov, moramo poskrbeti tudi za evidentiranje količinskega in vrednostnega zmanjšanja stanja, kar poskrbimo z entitetnima tipoma Narocilo in Dobavnica. V prvem primeru gre za količinsko zmanjšanje (ko dobimo naročilo, je izdelek rezerviran), v drugem pa za vrednostno zmanjšanje stanja zalog končnih proizvodov (ko dejansko dobavimo izdelek kupcem in s tem ustvarimo pogoje za obračun stroškov prodanih proizvodov). Ločevanje med vrednostnim in količinskim zmanjšanjem stanja zalog končnih proizvodov je primerno zaradi dejstva, da naročen izdelek še ne pomeni, da imamo na zalogi dejansko manj tega izdelka, je pa ta podatek pomemben pri sprejemanju novih naročil, ker tako lažje dobimo vpogled v dejansko stanje in dajemo realnejše obljube glede izpolnitve dobavnih rokov. S samimi proizvodi je povezan tudi entitetni tip Cenik, s pomočjo katerega so zabeležene vse cene proizvodov in storitev, klasificirane po kupcih glede na državo itd. Podjetje se namreč poslužuje segmentacije kupcev in različnim kupcev zaračunava različne cene (zato 1:N povezava), odvisno ali gre za domač ali tuji trg, EU ali NAFTA in drugi kriteriji.

Entitetni tip Delavec je poleg povezave z entitetnim tipom Potni nalog povezan na eni strani tudi z entitetnim tipom Delovni nalog in na drugi z entitetnim tipom Proizvodni delovni nalog. V

vseh primerih gre za M:N povezavo, saj je zaposlen kot nosilec aktivnosti lahko povezan z več delovnimi, proizvodnimi in potnimi nalogi in obratno, na vsakem od nalogov se lahko nahaja tudi po več zaposlenih hkrati.

Iz Slike 12 (na strani 35) je moč razbrati še štiri entitetne tipe, in sicer Obračun DN in DN nadurno delo, ki služita za obračun stroškov dela obdelave materiala (in s tem dodane vrednosti), Obračun PDN, ki služi za obračun stroškov končnih izdelkov, in Obračun PN, ki je povezan s samimi potnimi nalogi (dnevnice). Podatke o vseh teh štirih entitetnih tipih potrebuje računovodstvo, ki preko ustreznih kontov (entitetni tip Kontni plan iz Slike 11 na strani 34) skrbi za ažurnost glavne knjige.

5.3. RELACIJSKA SHEMA

Za transformacijo globalnega podatkovnega modela v relacijski model, ki ga bom predstavil z relacijsko shemo, sem se odločil zato, ker so v obravnavanem podjetju zaposleni ljudje, ki imajo izkušnje z Microsoftovim orodjem za upravljanje z relacijskimi bazami Access.

Pri načrtovanju podatkovne baze nas pogosto zanima le struktura relacije (in ne celotna tabela z vsemi podatki), ki je dobro opredeljena s svojim imenom in seznamom atributov. Tak zapis imenujemo relacijska shema, relacijska shema podatkovne baze pa bo praviloma vsebovala več takih shem za posamezne relacije. Relacijska shema podatkovne baze je hkrati njen načrt, saj opredeljuje, kako naj strukturiramo podatke: tabele, njihove attribute, glavne ključe itd. Seveda je potrebna še natančnejša opredelitev fizičnih značilnosti, na primer tipe in velikosti posameznih atributov, s tako imenovanim fizičnim načrtom.

Vsaka izmed entitet na strani 34 prikazanem globalnem podatkovnem modelu predstavlja relacijo, torej tabelo v relacijski podatkovni bazi. Povezave med entitetami so upoštevane s pomočjo atributov teh entitet (glavni, označeni s krepko pisavo, in tuji ključi, označeni z ležečo pisavo).

DOBAVNICA	(številka_dobavnice , datum_dobavnice, <i>šifra_naročila</i> , ime_prevoznika, Reg_št_vozila, neto_teža_skupaj, bruto_teža_skupaj)
NAROČILO	(šifra_naročila , datum_naročila, <i>šifra_kupca</i>)
KUPEC	(šifra_kupca , ime_kupca, naslov_kupca, kraj_kupca, tel_kupca, kontaktna_oseba, email)
PROIZVOD	(šifra_proizvoda , ime_proizvoda, enota_mere, kosovnica)
RAČUN	(številka_računa , datum_računa, znesek_računa, <i>številka_dobavnice</i> ,

	<i>šifra_kupca</i>)
CENIK	(vrsta_cene , šifra_proizvoda , datum_začetka, datum_zaključka, cena_enote)
DOBAVITELJ	(šifra_dobavitelja , ime_dobav, naslov_dobav, kraj_dobav, tel_dobav, kontaktna_oseba, email)
NAROČILO_DOB	(številka_naročila_dob , datum_naročila, <i>šifra_dobavitelja</i>)
PREVZEMNICA	(številka_prevzemnice , datum_prevzemnice, <i>številka_DN</i> , šifra_dobavnica_dob)
IZDAJNICA	(številka_izdajnice , datum_izdajnice, <i>številka_DN</i> , <i>številka_PDN</i> , šifra_dobavnica_dob)
DELOVNI NALOG	(številka_DN , datum_DN)
PROIZVODNI DELOVNI NALOG	(številka_PDN , datum_PDN)
DELAVEC	(šifra_delavca , ime_delavca, priimek_delavca)
PREDAJNICA	(številka_predajnice , datum_predajnice, <i>šifra_proizvoda</i> , <i>šifra_PDN</i> , količina)
MATERIAL	(šifra_materiala , naziv_materiala, merska_enota)
POTNI NALOG	(številka_PN , datum_PN, odh_čas, prih_čas, opombe)

Sledeče relacijske sheme se nanašajo na M:N povezave med entitetnimi tipi, ki jih predstavimo z uvedbo nove tabele (relacije). Ključi so sestavljeni iz ključev zgoraj definiranih zapisov.

NAROČILO_POST	(številka_naročila , šifra_proizvoda , količina_naročila)
DOBAV_POST	(številka_dobavnice , šifra_proizvoda , količina_dobav)
PRO_MAT	(šifra_proizvoda , šifra_materiala)
MAT_NAR_DOB	(šifra_materiala , številka_naročila_dob , količina_nar_dob)
PREVZ_MAT	(številka_prevzemnice , šifra_materiala , količina_prevz)

IZDAJ_MAT (številka_izdajnice, šifra_materiala, količina_izdaj)

DN_DELAVEC (številka DN, šifra_delavca)

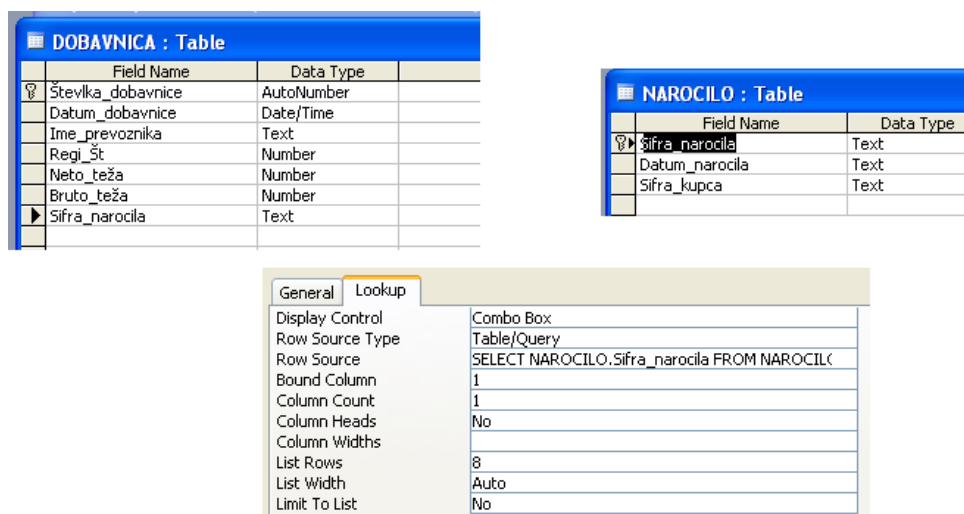
PDN_DELAVEC (številka PDN, šifra_delavca)

PN_DELAVEC (številka_PN, šifra_delavca)

5.4. PROTOTIP PREDLAGANE PODATKOVNE BAZE

Navedene sheme iz prejšnje točke zdaj enostavno pretvorimo v tabele v relacijski podatkovni bazi, za vsakega izmed atributov določimo tip, velikost znakov in druge omejitve oziroma značilnosti ter morebitne tuje ključe.

Slika 13: Definiranje tipov podatkov in povezave med entitetnima tipoma Dobavnica in Narocilo



Vir: Lastna opazovanja.

Iz Slike 13 so razvidni atributi entitenege tipa (v nadaljevanju tabele) Dobavnica in Narocilo ter sama implementacija povezave med njima. Ker je osnova za vsako dobavnico naročilo (naročilo torej zagotovo obstaja), lahko v orodju Access pod atributom Sifra_narocila tabele Dobavnica, shranimo poizvedbo, ki prebere vse šifre (ali kako drugače, odvisno od kriterija) naročil in torej onemogoča morebitne napake pri vnosu šifer naročil. Uporabljeni SQL stavek, ki se ga enostavno zapiše pod Look up opcijo obravnavanega atributa:

```
SELECT NAROCILO.Sifra_narocila FROM NAROCILO
```

S pomočjo opcije Switchboard Manager v programu Access lahko izdelamo uporabnikom prijazen vmesnik, preko katerega so v interakciji s podatkovno bazo. Na Sliki 14 je primer takega vmesnika.

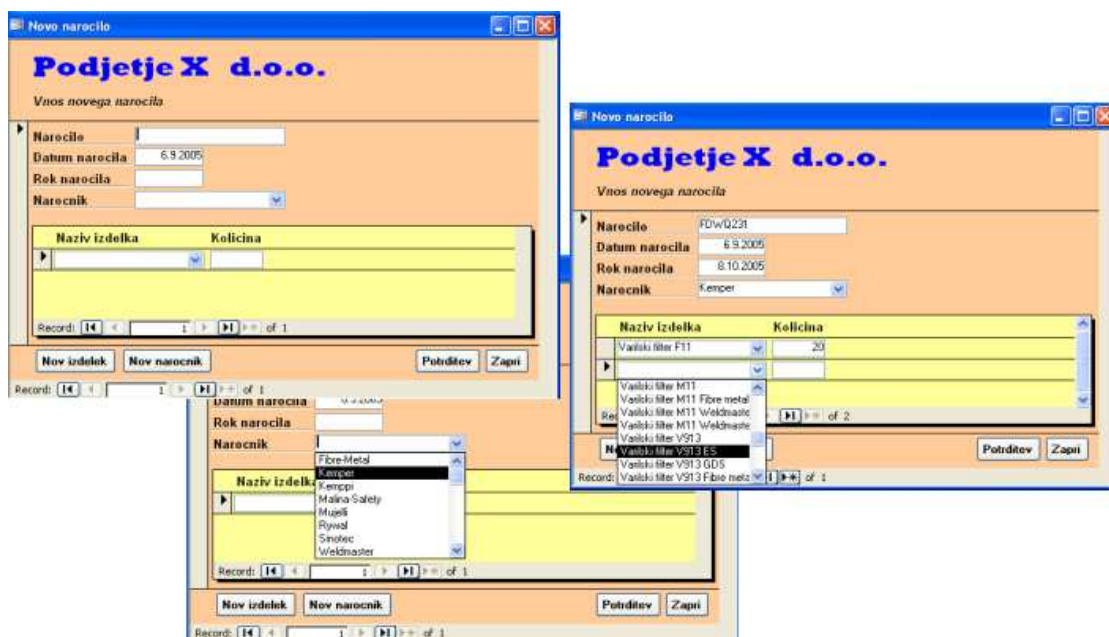
Slika 14: Prikaz menijev izdelanih s pomočjo Switchboard Manager opcije



Vir: Lastna opazovanja.

Iz Slike 14 je razviden seznam možnosti, ki uporabnika preko pritiska na ustrezen gumb pelje do ustreznih vnosnih obrazcev, poročil ali poizvedb. Primer takega obrazca za vnos novih naročil je na Sliki 15.

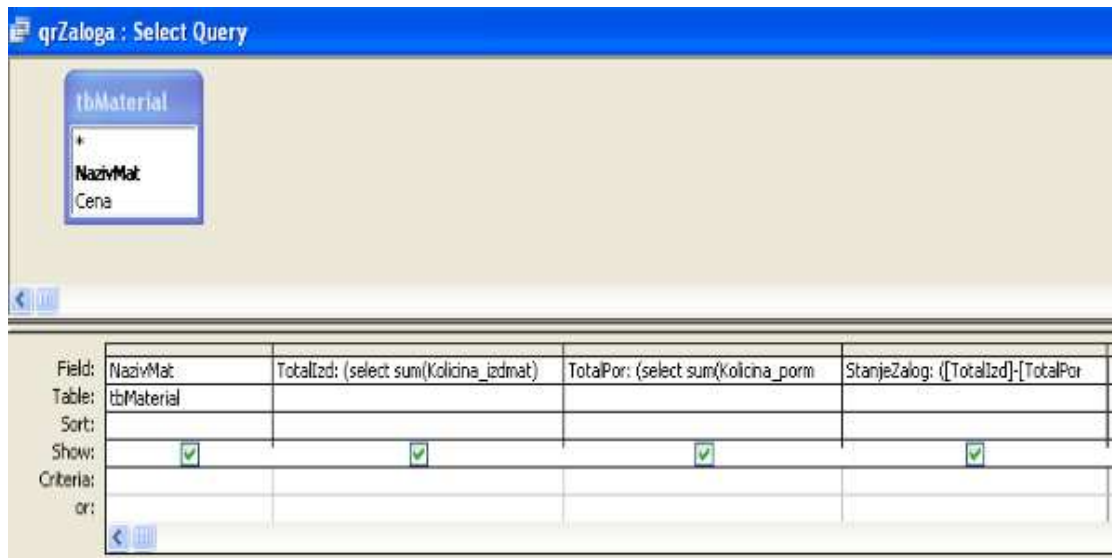
Slika 15: Prikaz obrazca za vnos novih naročil



Vir: Lastna opazovanja.

Iz Slike 15 (na strani 41) je razvidno, da polje Narocnik na obrazcu Novo narocilo vsebuje isti SQL stavek, kot v primeru iz Slike 13 (na strani 40), le da tokrat pregledujemo po vrednostih atributa v tabeli Kupec. Isto velja naprej za proizvode. Na ta način onemogočimo morebitne napake pri vnosu novih naročil.

Slika 16: Primer poizvedbe po stanju zalog



Vir: Lastna opazovanja.

Na Sliki 16 je primer poizvedbe, ki poišče seštevek količin izdanega materiala po posameznem materialu, seštevek količin prevzetega materiala po posameznem materialu in razliko med tema dvema, torej stanje zalog v določenem trenutku. Uporabljeni SQL stavki so:

TotalIzd:

(select sum(Kolicina_izdmat) from tbIzdajMat where Sifra_Mat = tbMaterial.Sifra_Mat)

TotalPor:

(select sum(Kolicina_pormat) from tbPrevzMat where Sifra_Mat = tbMaterial.Sifra_Mat)

StanjeZalog:

([TotalIzd]-[TotalPor])

Na podlagi te poizvedbe se lahko generira tudi poročilo, v priročnem A4 formatu, ali pa rezultate te poizvedbe uporabimo pri drugih poizvedbah, obrazcih, poročilih itd.

5.5. POSLEDICE UPOŠTEVANJA IZSLEDKOV OBRAVNAVANE TEMATIKE

Z uvedbo integrirane podatkovne baze, kot je prikazana v prejšnji točki, si podjetje lahko obeta veliko mero organiziranosti v smislu zajemanja podatkov v trenutku njihovega nastanka. Dobi vpogled v vsa aktivna naročila, lahko sledi materialu z različnih vidikov, kdo je bil delavec, ki ga je uporabljal, v zvezi s katerim delovnim nalogom in podobno. Poleg možnosti vzdrževanja evidenc o poslovnih partnerjih, dokumentiranja in spremljanja aktivnosti, jim taka podatkovna baza omogoča tudi samo standardizacijo postopkov in podatkov, ki se zajemajo na eni strani in posredujejo naprej v računovodstvo na drugi.

Poleg prednosti je potrebno omeniti tudi druge posledice uvedbe prototipa podatkovne baze. Vsaj za obravnavano podjetje uvedba tega prototipa predstavlja problem, ker enostavno nima zaposlenih, ki bi se ukvarjali s pisarniškim delom ali kogarkoli, ki bi bil odgovoren za vnos podatkov v podatkovno bazo. V marsikaterem primeru tudi ni možno, da bi to opravljala le ena oseba, kar pomeni dodatne odgovornosti zaposlenih oziroma dodatni napor z vidika priučitve in podobno. V vsakem primeru pride do nekih dodatnih stroškov, ki so posledica organizacijskih naporov in sprememb v samih postopkih dela.

Zelo rado se namreč zgodi, da se v primeru preveč temeljnih sprememb, v korelaciji z izobrazbeno strukturo zaposlenih ali katerimi drugimi dejavniki, podjetja potemtakem raje ne odločajo za uvedbo takih radikalnih sprememb čez noč. Kljub temu sem mnenja, da bi prednosti, ki izhajajo iz uvedbe integrirane podatkovne baze, na dolgi rok odtehtale vse morebitne slabosti povezane z njeno uvedbo. Res je, marsikatero izmed prednosti se ne da ovrednotiti s stroškovnim prihrankom, vendar je potrebno upoštevati tudi druge kriterije, kot so sistematičnost, organiziranost, pravočasnost, hitrost, zanesljivost in podobno.

6. SKLEP

Za vsako podjetje je izjemnega pomena razviti in uporabljati informacijski sistem, ki bo zmožen odgovarjati na informacijske potrebe organizacije tako danes kot v bodoče. Pri samem postopku izgradnje informacijskega sistema je zato uporaba dognanj s področja managementa podatkov ne le dobrodošla, ampak zaradi razvoja računalniške in komunikacijske tehnologije ter same dinamike poslovnega okolja organizacije tudi nujna. Z uvedbo integrirane podatkovne baze, ki temelji na podrobni poslovni analizi organizacije, njenih informacijskih potreb in značilnosti ter predhodno izdelanim modelom poslovnih procesov in modelom podatkov, podjetje zadosti vsem merilom, ki zagotavljajo dobra izhodišča za nastanek jedra informacijskega sistema.. Ta bo hkrati temeljil na upoštevanju tako potreb in zahtev različnih uporabnikov podatkovne baze kot zahtev in možnosti danes dostopne tehnologije na področju podatkovnih baz in sistemov za upravljanje s podatkovnimi

bazami. Organizacija, ki je odločena postati in ostati prožna in odzivna na dolgi rok, mora ta spoznanja upoštevati ter jih vpeti v njihov poizkus modeliranja njim lastnih aktivnosti, procesov in podatkov ter na ta način spoznavati samega sebe, svojih potreb in svoje vloge v širšem, družbenem merilu.

LITERATURA

1. Brathwaite S. Kenmore: Relational Theory: Concepts and Application. San Diego : Academic Press, 1991. 261 str.
2. Domajnko Tomaž, Heričko Marjan, Rozman Ivan: Uporaba objektnih podatkovnih baz. Zbornik posvetovanja: Dnevi slovenske informatike 1997. Ljubljana : Slovensko društvo informatika, 1997, str. 183-191.
3. Grad Janez, Kovačič Andrej, Barle Janez: Podatkovne strukture, osnove baze podatkov in njene uporabe. II. del. Ljubljana : Ekonomska fakulteta, 1990. 151 str.
4. Grad Janez, Kovačič Andrej, Barle Janez: Osnove baze podatkov in njene uporabe. I. in II. del. Ljubljana : Ekonomska fakulteta, 1996. 120 str.
5. Gradišar Miro, Resinovič Gortan: Informatika v poslovnem okolju. Ljubljana : Ekonomska fakulteta, 1996. 479 str.
6. Komel Janez: Približevanje relacijskih in objektnih baz. Zbornik posvetovanja: Dnevi slovenske informatike 1998. Ljubljana : Slovensko društvo informatika, 1998, str. 19-26.
7. Kovačič Andrej et al.: Prenova in informatizacija poslovanja. Ljubljana : Ekonomska fakulteta, 2004. 345 str.
8. Laudon C. Kenneth, Laudon P. Jane: Management Information Systems. Second Edition. New York : Macmillan Publishing Company, 1991. 940 str.
9. Laudon C. Kenneth, Laudon P. Jane: Management Information Systems. Sixth Edition. New Jersey : Prentice-Hall International, 2000. 588 str.
10. Lucas C. Henry, Jr: The Analyses, Design, and Implementation of Information Systems. Singapore : McGraw-Hill, 1992. 550 str.
11. McFadden R. Fred, Hoffer A. Jeffrey: Data base management. Second Edition. Menlo Park : The Benjamin/Cummings Publishing Company, 1988. 680 str.
12. McLeod Raymond Jr., Schell George: Management Information Systems. Eighth Edition. New Jersey : Prentice-Hall International, 2001. 478 str.
13. Špehar Katja: Podatkovno modeliranje na primeru paketne programske rešitve. Diplomsko delo. Ljubljana : Ekonomska fakulteta, 2004. 42 str.
14. Tkalac Slavko: Relacijski model podatka. Zagreb : IRO "Informator", 1988. 152 str.
15. Vintar Mirko: Konceptualno modeliranje informacijskih sistemov. Doktorska disertacija. Ljubljana : Ekonomska fakulteta, 1990. 159 str.
16. Vožič Bogdan: Metodologija zasnove baze podatkov. Magistrsko delo. Ljubljana : Ekonomska fakulteta, 1991. 142 str.

VIRI

1. Database Models [URL: <http://unixspace.com/context/databases.html>], 21.7.2005.
2. Interno gradivo podjetja. Ljubljana, 2005.