

MODELI RAZVOJA PROSTORSKEGA INFORMACIJSKEGA SISTEMA

dr. Radoš Šumrada
FAGG-Oddelek za geodezijo, Ljubljana
Prispelo za objavo: 21.1.1994

Povzetek

Razvoj informacijskega sistema obsega številne zaporedne razvojne stopnje, od katerih vsaka tvori podrobnejšo in bolj konkretno nadgradnjo vseh predhodnih dejavnosti. Sistemski razvoj se tako lahko interpretira kot sestavljeni proces izdelave podrobnih sistemskih modelov. Pod pojmom sistem se ne pojmuje samo softverski sistem. Takšen sistem je lahko tudi celoten poslovni ali prostorski informacijski sistem, katerega sestavni deli so strojna oprema, komunikacijsko omrežje, organizacijski postopki, sistemska ter uporabniška programska oprema in podatki. V članku so predvsem značilnosti strukturnega in objektno orientiranega modeliranja informacijskega sistema.

Ključne besede: CASE, načrtovanje, objektno orientirana analiza, razvoj informacijskega sistema, strukturna analiza

Abstract

A development of an information system implies numerous consecutive developmental stages where each forms a more detailed scaling of all former activities. System development may thus be interpreted as a connected process of detailed system models production. The term system does not comprise only a software system. Such a system may be a large business or spatial information system composed of hardware, communication network, organisational procedures, system and application software, and data. The paper outlines above all characteristics of a structural and object-oriented modelling of an information system.

Keywords: CASE, design, information system development, object oriented analysis, structured analysis

1. UVOD

Proces načrtovanja in razvoja informacijskih sistemov je drag, dolgotrajen, dinamičen ter ponavljajoč se postopek. Razvoj vsakega sistema je proces zaporednih sprememb obstoječega sistema zaradi novih in spremenjenih uporabniških zahtev. Nobena filozofija ali obstoječi metodološki niz principov ne more v celoti zadovoljiti vseh zahtev, ki se pojavljajo v takšnih postopkih. Razvojni

koncepti informacijskih sistemov se lahko danes manifestirajo, čeprav z različnimi poudarki in pristopi, kot tri najbolj prominentne razvojne metodološke šole, ki na področju razvoja informacijskih in softverskih sistemov neprestano tekmujejo za vodilno vlogo. Dodatno skupino predstavljajo razne individualne metode, ki so se posamično razvile in se uporabljajo samo v posameznih okoljih (Yourdon 1992). Obstoječe metode za razvoj sistemov se lahko razdelijo na štiri metodološke skupine: različne individualne metode za podatkovno in procesno modeliranje, strukturno modeliranje, informacijsko inženirstvo, objektno orientirane metode.

Celotni tehnološki in proceduralni načrt za razvoj ter realizacijo novega informacijskega sistema se imenuje njegov razvojno življenjski cikel (Yourdon 1988). Tvorijo ga naslednje razvojne faze: strateško planiranje, systemska analiza, systemsko načrtovanje, konstrukcija ter izvedba sistema, realizacija in operativno delovanje sistema. Po analogiji z živimi bitji lahko razvojno-življenjski cikel informacijskega sistema pregledno razdelimo na naslednje obdobja :

- obdobje načrtovanja, razvoja in ustvarjanja informacijskega sistema
- obdobje delovanja, vzdrževanja in nadaljnega razvoja sistema
- obdobje zastaranja in opustitve informacijskega sistema.

2. RAZVOJNA METODOLOGIJA, TEHNIKE IN ORODJA

Celoten sklop in zaporedje dejavnosti, ki so potrebne za razvoj, realizacijo ter delovanje nekega informacijskega sistema, se imenuje njegov razvojno-življenjski cikel. Postopki ter aktivnosti v takšnem sklopu se običajno izvajajo v skladu z dobro opredeljenimi in doslednimi postopki, ki temeljijo na izbrani metodologiji. Uporabljena metodologija vsebuje specifične dejavnosti za analizo in načrtovanje logičnega podatkovnega modela, fizične podatkovne strukture kakor tudi procesnega modela oziroma operativnega dela sistema. Za podporo vsaki metodologiji so potrebna ustrezna sredstva za njeno realizacijo. To so orodja za podporo in organizacijske tehnike za njeno izvedbo.

Običajno je, da se pri razvoju zahtevnih informacijskih sistemov uporabljajo številne tehnike. Vsaka uporabljena razvojna tehnika prikazuje različne poglede na celoten sistem in poudarja nekatere izbrane vidike sistema ter zapostavlja ostale. Tipična razvojna metoda zato uporablja številne tehnike, tako da pokriva mnoge vidike systemske analize in načrtovanja. Razvojne tehnike večinoma tudi niso med seboj izključujoče in vezane na eno samo metodo. Enaka tehnika se lahko uporablja v številnih razvojnih metodah. Čeprav je določena razvojna tehnika enaka, ima vsak vpliven razvojni metodolog tudi navado, da spremeni in po potrebi prilagaja splošno tehniko. Tako nastajajo novi razvojni in modelni pomeni in pojavljajo se zlasti nove grafične notacije. Pri vsaki razvojni tehniki sta dva ključna vidika, ki ju je treba posebej poudariti. To sta strogost tehnike in njena razumljivost. Stroga tehnika je predvsem dosledna tehnika, ki ne dovoljuje dvoumnih interpretacij. Strogost razvojne tehnike je zelo pomembna kvaliteta, ker se lahko sicer številne dvoumnosti in nedoslednosti modelov prenesejo vse do same izvedbene faze sistema.

Razumljivost razvojne tehnike je često premalo poudarjena, vendar pa je tudi izredno pomembna. Razvojna tehnika, ki je težko razumljiva, je predvsem zahtevna za uporabo in se jo težko poučuje. Takšna zahtevna razvojna tehnika tudi

zmanjšuje ponovno uporabo že razvitih sistemskih modelov, ki postaja ključni element tudi za ponovno uporabo že razvite programske opreme. Dodaten razlog, ki govori v korist uporabe lahko razumljivih razvojnih tehnik je v tem, da morajo imeti sistemski analitiki in načrtovalci dober pregled nad sistemskimi modeli. Če je določen model težko razumljiv, je zelo verjetno, da bo neposredni rezultat tudi manj kvalitetni končni model sistema. Strogost in razumljivost vsekakor nista vzajemno izključujoča vidika. Grafični diagrami so lahko zelo pregledni in razumljivi ter se lahko zelo dosledno opredelijo, tako da postanejo popolnoma formalni. Vsaka razvojna tehnika poudarja določene vidike sistema in predstavlja značilen pogled na sistem. V splošnem obstajajo trije osnovni pogledi na sistem, ki se kažejo tudi kot trije dopolnjujoči se modeli sistema:

- podatkovni model prikazuje predvsem vsebino in značilnosti problemskega področja
- model systemskega obnašanja ponazarja dinamiko sistema in mora zagotavljati različne systemske odgovornosti
- procesni model je pogojen z osnovno funkcionalnostjo sistema in mora odražati predvsem uporabniške zahteve.

Enostavni primer razvojnega modela izdelave informacijskega sistema temelji na definiciji in predstavitvi njegovih zunanjih lastnosti, ki jih tvorijo procesni, dinamični in podatkovni model ter systemske poslovno organizacijske značilnosti. Predstavljeni razvojni model predpostavlja CASE metodološki pristop in uporabo orodij CASE za avtomatizacijo postopkov, čeprav to ni izrecno poudarjeno (Kratice CASE se v računalništvu pojmuje na dva načina. Pomen kratice CASE se v angleščini lahko interpretira v ožjem in širšem pomenu, ki sta naslednja: – Computer Aided Software Engineering, – Computer Aided Systems Engineering). Orodje CASE so posebna računalniška programska oprema ter ostali pripomočki za avtomatizirano izdelavo podatkovnega modela, systemske programske opreme in potrebnih uporabniških aplikacij. Opisani primer temelji na tradicionalnem evolucijskem ali zaporednem razvojnem modelu, pristopu od zgoraj navzdol, na uporabi strukturne analize in načrtovanja (Yourdon 1989) ter nekaterih diagramskih tehnik informacijskega inženirstva. Nakazane so tudi osnovne razvojne razlike pri uporabi objektivno orientirane analize in načrtovanja.

3. PRIMER STRUKTURNEGA RAZVOJA INFORMACIJSKEGA SISTEMA

Razvojni projekt informacijskega sistema se običajno začne s predlogom ali izraženo željo o novih ali spremenjenih informacijskih zahtevah (Barker 1990b). Zamisel za razvojni projekt izhaja ponavadi iz okolja vodilnih poslovnih upravljalcev obstoječega sistema. Registrirane nove uporabniške zahteve so običajno rezultat izvedenega strateškega planiranja. Ta podaja uvodni pregled ter prvo analizo spremenjenih informacijskih, organizacijskih, poslovnih, tehnoloških ali političnih okoliščin, ki predstavljajo novo razvojno in poslovno priložnost. Rezultati strateškega plana so podani kot poslovna vizija, razvojni načrti in arhitektura novega sistema. Običajno so zelo neformalni in splošni. Analizirani problemi, ki jih je treba rešiti, so običajno podani kot strateški razvojni cilji, poslovni načrti in omejitve. Osnovni poudarek v strukturnem metodološkem pristopu je predvsem na funkcionalnem

oziroma procesnem modeliranju sistema in kasnejši posredni izpeljavi pogosto ločenega podatkovnega modela. Formira se skupina strokovnjakov za sistemsko inženirstvo in informatiko, ki mora najprej podrobneje analizirati in formulirati uporabniške probleme ter zahteve. Definirati je treba značilnosti problemskega področja, vse uporabniške zahteve in probleme, ki jih je treba opisati s podatkovnega, procesnega ter organizacijskega vidika. Na podlagi podanih problemov in omejitev se izdelata okvirni opis možnih rešitev. Upošteva se zlasti vpliv možnih rešitev na organizacijsko zgradbo sistema in sprejemljivost predlaganih rešitev glede na vse postavljene cilje.

Na samem začetku faze strateškega planiranja se izdelata tudi analiza stroškov in koristi ter študija izvedljivosti projekta. Nato se običajno na podlagi rezultatov analize stroškov in koristi poda podroben pregled alternativnih razvojnih možnosti ter možnih organizacijskih arhitektur sistema. Kot rezultat celotne faze strateškega planiranja se izdelata predvsem uvodna analiza stroškov in koristi, ki podaja naslednje značilnosti načrtovanega informacijskega sistema:

- oceno direktnih in posrednih stroškov (razvojni stroški za strojno, celotno programsko in komunikacijsko opremo, šolanje potrebnih kadrov, zagon ter letno vzdrževanje sistema),
- oceno opredeljivih in posrednih koristi od novega sistema (informacijske izboljšave, povečana produktivnost in učinkovitost, izboljšana ekonomija ter poslovanje, zmanjšani stroški, posredno izboljšana zanesljivost odločanja in druge neopredeljive koristi),
- relativna primerjalna analiza stroškov in koristi med obstoječim ter novim informacijskim sistemom za večletno obdobje (tehnične ter organizacijske izboljšave, vrednost investicije, analiza dotoka in pretoka sredstev, ocena povrnitve stroškov in dobička).

Sledi analitična faza, v kateri se mora podrobno analizirati problemsko področje, vse uporabniške zahteve, sistemske odgovornosti, razčleniti in pojasniti vse probleme ter podati podrobno analizo danosti. Analiza je v bistvu proces razčlenitve pomenke ali materialne celote na njene sestavne dele zaradi njihove proučitve. V tej fazi sistemski analitiki razvijejo celotni sklop sistemskih zahtev s pomočjo analize obstoječega podatkovnega in procesnega modela sistema. Za ponazoritev modela sistema se uporabijo različne diagramске tehnike strukturne analize. Osnovni cilj analitične faze je, da se predvsem podrobno prouči obstoječe stanje in opredelijo vsi problemi. Izdelata se podrobna specifikacija lastnosti in funkcionalnosti sistema, ki jih podajajo procesni, dinamični ter podatkovni model sistema. Dejanski rezultat strukturne analize je izrazito mrežna analitična predstavitev celotnega sistema.

Okvirni pregled celotnega procesnega modela sistema in podsistemov podaja kontekstni diagram, ki se nato razstavi na vedno več nižjih podrobnejših procesnih nivojev (Barker, Longman 1992). Uporabijo se diagrami pretoka podatkov (DFD), ki podajajo procese in pretok podatkov v obstoječem sistemu ter ponazarjajo, kaj je treba narediti za alternativno in izboljšano reševanje problemov. Diagrami pretoka podatkov prikazujejo meje sistema, načine vstopa podatkov v sistem, pretok podatkov po sistemu, kakšne podatke določene funkcije in aktivnosti potrebujejo ter

kako jih preoblikujejo, način hranjenja ključnih podatkov v sistemu in možne načine izstopa podatkov iz sistema.

Model sistemskega procesnega obnašanja ali dinamični model sistema opredeljuje spremembe sistemskih stanj (Shlaer, Mellor 1992). Sistem je zmeraj v enem od svojih stanj in je lahko hkrati samo v enem od svojih stanj. Ko se določen za sistem pomemben dogodek pripeti, mora sistem nanj reagirati v skladu z aktivnostmi, ki jih trenutno opravlja. Pomen dinamičnega modela sistema je odvisen predvsem od vrste informacijskega sistema (registrativni ali v realnem času), ki se modelira, vendar pa ga je treba zmeraj vsaj proučiti. Model sistemskega obnašanja prikazuje zaporedje aktivnosti in dogodkov, ki lahko spreminjajo statično upodobitev podatkovnega modela sistema. Glavna grafična tehnika za dinamično modeliranje sistema so postali tako pri strukturalni razvojni metodi kakor tudi v večini objektno orientiranih razvojnih metod diagrami prehodnih stanj (STD). Ta oblika diagrama ima številne prednosti, ki lahko ponazorijo celotno sistemsko dinamiko. Prikazuje vsa pomembna stanja sistema, ki jih povezujejo vsi možni prehodi med sistemskimi stanji ter dogodki, ki takšne spremembe neposredno povzročajo. Za določitev podatkovnega modela sistema in načine shranjevanja podatkov na datotekah ali v podatkovni bazi se običajno razvijejo entitetno relacijski diagrami (Barker 1990a). Entitetno relacijski diagrami (ERD) običajno predstavljajo tudi osnovno začetno vsebino podatkovnega slovarja CASE. V podatkovni slovar se shranjujejo vsi podatki o podatkovni strukturi, vsi opisi procesov in aktivnosti v sistemu ter pregledne presečne podatkovno procesne matrike za ponazoritev in kontrolo procesno podatkovnega modela sistema. Sistemski inženirji nato pripravijo celovit pregled o izpolnitvi uporabniških zahtev, ki se dodatno podrobno razčlenijo. Izbere se osnovna podatkovna in procesna arhitektura ter označijo še sprejemljive alternativne možnosti. Na koncu analitične faze se izdelata tudi podrobnejša analiza predvidenih kvalitativnih razvojnih stroškov in koristi.

Sledi faza sistemskega načrtovanja, v kateri sistemski analitiki predajo vse deloblasti sistemskim načrtovalcem in programerjem, ki izdelajo načrt za logično podatkovno ter procesno zgradbo vseh komponent sistema. V fazi načrtovanja se že predvidevajo in morajo upoštevati načrtovane softverske in hardverske značilnosti bodočega informacijskega sistema. Osnovna strategija strukturnega načrtovanja temelji na določitvi vseh elementarnih procesnih korakov, ki so potrebni za nov informacijski sistem. Osrednji razvojni poudarek je torej na principih in načinu določitve potrebnih procesnih lastnosti novega sistema. Detajlni diagrami pretoka podatkov se pretvorijo v procesne in kontrolne module. Izdelajo se strukturne karte (SC), ki prikazujejo hierarhijo vseh sistemskih funkcij. Sistemske funkcije na najvišjem nivoju so tako imenovane kontrolne in vodilne funkcije, ki vodijo procesiranje v sistemu ter kličejo različne izvedbene funkcije na nižjih nivojih. Detajlni entitetno relacijski diagrami se predelajo v logično podatkovno zgradbo za potrebe izbranega tipa podatkovne baze.

Cilj faze načrtovanja in izvedbe so fizična struktura sistema, ustreznost programske kode v izbranem strukturnem programskem jeziku ali/in struktura relacijske podatkovne baze v smislu normaliziranih tabel. Rezultat je hierarhična razčlenitev celotnega modela sistema na procesne dinamične funkcije, katerim se podajajo potrebni pasivni podatki. V tradicionalni fazi izgradnje informacijskega sistema se

nato izdelava podrobna fizična struktura in specifikacija podatkovne baze na podlagi entitetno relacijskih diagramov. Sledi sestava uporabniških programov in aplikacij, kjer se dejansko kodirajo procesne aktivnosti oziroma metode razredov. Programerji oziroma često tudi sistemski načrtovalci sestavijo programsko kodo na podlagi podrobnih načrtov. Detajlni diagrami pretoka podatkov, strukturne karte in diagrami prehodnih stanj so osnova za sestavo programske kode za vse funkcije in rutine. V sodobnih orodjih CASE se potrebna programska oprema danes že sorazmerno avtomatizirano sestavlja s pomočjo ustreznih aplikacijskih in programskih generatorjev.

4. ENOSTAVNI PRIMER OBJEKTNO ORIENTIRANEGA RAZVOJA INFORMACIJSKEGA SISTEMA

Sistemsko modeliranje sestavlja niz prepletenih aktivnosti za zaporedno izdelavo povezanih, vedno bolj progresivnih in vedno bolj podrobnih modelov sistema. Sistemsko analizo in načrtovanje sta dva jedrnata pristopa za opisovanje ter modeliranje sistema. Sistemsko analizo je študij in modeliranje določenega problemskega področja v smislu zadovoljevanja uporabniških zahtev in zagotavljanja sistemskih odgovornosti. Osnovni cilj objektno orientirane analize je podrobna opredelitev in opis sistema v smislu, kaj je treba storiti za rešitev določenega systemskega problema ter spremenjenih uporabniških zahtev. Objektno orientirana analiza predstavlja modeliranje sistema brez upoštevanja značilnosti njegovega dejanskega izvedbenega okolja. Sistemsko načrtovanje je niz aktivnosti, ki na podlagi analitičnih dognanj opredeljuje načine zajemanja, vzdrževanja, procesiranja, dostopa, razdeljevanja in predstavitve podatkov v namenskem sistemu. Glavni cilj objektno orientiranega načrtovanja je v podrobni specifikaciji v smislu, kako je treba izvesti in realizirati analitično opredeljen sistemski problem. Objektno orientirano načrtovanje pomeni progresivno razširitev modelov objektno orientirane analize ter že vključuje odločitev o določeni sistemski izgradnji in arhitekturi glede na učinkovitost ter ponovno uporabo že razvitih systemskih modulov. Osnovni cilj objektno orientiranega načrtovanja je podrobna opredelitev notranje vsebine razredov, ki vključuje podrobno navedbo vseh atributov, vmesnikov, algoritmov in metod razredov.

V primeru objektno orientiranega razvoja se analitična faza in načrtovanje izvedeta precej drugače. Najprej se celotno problemsko področje razčleni na obvladljiva področja ali podsisteme. Sledi identifikacija in razčlenitev izbranih stvarnih entitet v potrebne vsebinske objekte, ki predstavljajo abstrakcijo pojavov problemskega področja. Določijo se tudi vsi glavni atributi objektov. Nato se objekti klasificirajo v tipološke objektno tipe, ki predstavljajo meta model objektov z enakimi atributi in procesnimi lastnostmi. Sledi potrebna klasifikacija objektnih tipov v ustrezno hierarhijo načelu generalizacije, specializacije in agregacije objektov v podin nadobjektno tipe. Določijo se tudi vse povezave oziroma dinamične relacije med objektnimi tipi, in sicer po tipu, načinu ali pogojnosti in stopnji ali kardinalnosti. Glavna diagramska tehnika za predstavitev in specifikacijo objektnih tipov ter njihovih povezav so objektni diagrami. Za dinamične objektno tipe in relacije je treba določiti tudi njihovo časovno spremenljivo obnašanje. Izdelava se tako imenovani model prehodnih stanj objektnih tipov in relacij, ki ponazarja njihov časovno urejeni

vzorec procesnega obnašanja. Na koncu se še identificira procesni model sistema, oziroma se okvirno opredelijo vse potrebne metode objektnih tipov, ki jih predstavljajo v objektnih združene procesne funkcije.

V objektno orientiranem pristopu predstavlja fazo načrtovanja podrobna določitev glavnih štirih logičnih komponent modela, ki so naslednje: celotno problemsko področje, organizacija podatkov, organizacija opravil in izdelava vmesnikov. Vsi rezultati objektno orientirane analize problemskega področja se privzamejo in izdelava se podroben izvedbeni načrt za dejansko realizacijo objektno orientiranega podatkovnega modela. Komponenta za organizacijo podatkov določa možne dostope in načine upravljanja s trajnimi objekti. Komponenta za organizacijo opravil predstavlja izdelavo izvedbenega načrta za določitev algoritmov in koordinacijo procesnih aktivnosti ter komunikacijskih potreb med objekti. Nato se določi pregledni načrt za organizacijo podatkovne baze. Zadnja komponenta v fazi načrtovanja služi za izdelavo različnih uporabniških programov in vmesnikov. V fazi izgradnje informacijskega sistema se izdelava detajlna fizična specifikacija podatkovne baze. Sledi sestava uporabniških programov in aplikacij, kjer se dejansko kodirajo procesne aktivnosti oziroma metode razredov. Programerji oziroma često tudi sistemski načrtovalci sestavijo programsko kodo na podlagi podrobnih načrtov. Detajlni objektni diagrami, diagrami pretoka podatkov in diagrami prehodnih stanj dinamičnih objektnih tipov so osnova za sestavo programske kode v izbranem objektno orientiranem programskem jeziku. V orodjih CASE je izdelava potrebne programske opreme že sorazmerno avtomatizirana in se sestavlja s pomočjo ustreznih aplikacijskih in programskih generatorjev. Sledita operativna faza in faza testiranja sistema, v katerih se celotni informacijski sistem podrobno preizkusi med dejanskim delovanjem. Verificirajo in preizkusijo se tudi operativne in izvedbene lastnosti programske opreme v skladu z uporabniškimi in funkcionalnimi zahtevami. Nato sledi običajno začasno vzporedno delovanje novega in starega informacijskega sistema. Novi sistem mora dokazati, da je sposoben zadovoljiti vse uporabniške zahteve. Po določenem obdobju, ki je potrebno za takšno testiranje, novi informacijski sistem nadomesti starega.

5. ZAKLJUČEK

Trenutno še ne obstaja nek splošen ter formaliziran pristop k izvedbi objektno orientirane systemske analize in načrtovanja. Uporablja pa se veliko splošnih pristopov, ki se lahko vsi opredelijo kot bolj ali manj hevristični pristopi. Najbolj pogost pristop k modeliranju informacijskega sistema je podatkovni pogled na sistem. Najprej se identificirajo objektni tipi, pomembni atributi, statične in dinamične strukture med objektnimi tipi ter osnovne procesne funkcije objektnih tipov. Celoten podatkovni model sistema se podrobno prikaže na ustreznem številu objektnih diagramov. Za prikaz procesnega obnašanja sistema se za vsak dinamičen objektni tip izdelava diagram prehodnih stanj, ki lahko prikazuje tudi posredovanje vseh vrst sporočil oziroma podatkov po sistemu. Osnovni problem pri takšnem pristopu je v tem, da je težko omejiti ustrezen obseg systemske analize. Težko se namreč določi najbolj primeren trenutek za prehod iz analize v načrtovanje sistema in njegovih izvedbenih razredov. Ta ugotovitev je še zlasti pomembna pri tradicionalnem strukturnem razvojnem pristopu. V vseh tradicionalnih strukturnih ter informacijskih

metodoloških razvojnih pristopih se vsebinski podatkovni model analitične faze močno razlikuje od logičnega in fizičnega modela, ki se uporabljata v fazah načrtovanja in izgradnje sistema. Sistemski analitik, načrtovalec in programer uporabljajo zelo različne podatkovne ter funkcijske modele sistema. Pri modeliranju uporabljajo analitiki zlasti entitetno relacijske diagrame, diagrame prehodnih stanj za ponazoritev dinamike in pregledne referenčne matrike za povezave. Načrtovalci imajo drugačen pogled na model sistema ter uporabljajo zelo podrobne diagrame pretoka podatkov, akcijske diagrame in strukturne karte za dekompozicijo vseh procesnih funkcij. Programerji uporabljajo spet drugačen vsebinski model in ne razmišljajo več v smislu diagramov pretoka podatkov ter strukturnih kart. Sistemski in aplikativni programerji uporabljajo prevajalnike, izdelane knjižnice ter avtomatske generatorje programske kode in uporabniške aplikacije. Relacijske podatkovne baze uporabljajo spet povsem drugačen vsebinski in fizični model sistema, ki temelji predvsem na normaliziranih tabelah. Načrtovanje podatkovnega modela relacijske podatkovne baze se zato zelo razlikuje od načrtovanja zgradbe programov. Aplikacijski programer mora sam ugotoviti, kako bo zajemal in shranjeval podatke v relacijske tabele.

V primerjavi s tradicionalnimi sistemskimi razvojnimi metodologijami je zelo pomembna prednost objektno orientirane metodologije predvsem v tem, da se v vseh ključnih razvojnih fazah sistema, ki so sistemski analiza, načrtovanje in izgradnja, uporabljajo skoraj popolnoma enaki modeli sistema. Strukturne metode na primer zahtevajo, da se izrazito mrežna analitična predstavitev sistema hitro pretvori v hierarhično več ravensko predstavitev sistema v fazi sistemskega načrtovanja. Objektno orientirano načrtovanje pa predstavlja naravno nadaljevanje objektno orientirane analize. Povezava ter prehod med objektno orientirano analizo in načrtovanjem sta mnogo bolj neposredna. Obstaja pa pomembna sprememba osrednjega žarišča, ki se prestavi iz določanja problemskega področja na področje reševanja in realizacije sistemskih modelov. Obstajata dva osnovna pristopa za prehod iz systemske analize k načrtovanju. Prvi pristop temelji na postopni razširitvi analitičnih modelov z dodatnimi elementi, ki so potrebni za dejansko načrtovanje sistema. Takšen načrtovalski pristop je načeloma končan, ko so razviti modeli, primerni za realizacijo ali programsko kodiranje. Težava pri takšnem pristopu je zlasti v tem, da nekatere obstoječe metode ne uporabljajo enake grafične notacije in oznake v vseh razvojnih fazah sistema. Določene objektno orientirane metode resnično uporabljajo zelo podobno grafično notacijo skozi celotni razvojni cikel sistema, vendar pa obstajajo tudi metode, ki namenoma zahtevajo spremembo uporabljenih oznak, tako da poudarjajo razliko med vsebinskim in fizičnim modelom sistema. Drugi kontrastni pristop za prehod iz systemske analize k načrtovanju temelji na transformaciji modelov. Takšni razvojni pristopi uporabljajo niz določenih pravil, ki služijo za ustrezno pretvorbo opredeljenih analitičnih modelov v ustrezne modele, ki so potrebni za načrtovanje sistema.

Poseben pristop v tej smeri je uravnavanje celotnega razvojnega procesa s pomočjo različnih možnih scenarijev oziroma predvidenih različnih načinov uporabe sistema od njegovih dejanskih uporabnikov. Takšen pristop je posebej podrobno obdelan v odlični svojevrstni razvojni tehniki Objectory (Jacobson 1992). Objectory je

obsežna objektno orientirana metodologija in proces za izdelavo velikih industrijskih informacijskih sistemov. Osnovni pristop je opredelitev različnih vrst uporabe sistema, ki pokrivajo vse možne sistemske namembnosti. Primer uporabe je določena oblika ali vzorec uporabniškega pristopa. Nek uporabnik sistema izvede določen scenarijo oziroma predviden način uporabe sistema. Uporabnik sproži nekaj transakcij ali pa zaporedje povezanih dogodkov. Proučevanje primerov uporabe se ne konča v fazah analize in načrtovanja, temveč se nadaljuje skozi tudi skozi fazi izvedbe in testiranja sistema. Povsem drugačen pristop uporablja (Martin, Odell 1993), ki temelji na modeliranju vseh za sistem pomembnih dogodkov. Takšen, na dogodkih temelječ pristop poudarja predvsem analizo vseh poslovnih procesov v sistemu, ki določajo obseg celotnega podatkovnega modela poslovnega sistema. Vsekakor pa nobena od do sedaj razvitih objektno orientiranih razvojnih tehnik ni brezhlebna in univerzalna. Za sistemskega analitika je torej pomembno tudi spoznanje, da je potrebno zelo pogosto ustrezno prikrojevanje posameznih razvojnih metod. Predhodno pa je seveda nujno potrebno dobro razumevanje značilnosti problemskega področja in uporabljene razvojne metodologije, preden je mogoče začeti z ustreznim prirejanjem izbranih metod. Razumevanje metodologije mora temeljiti na njenih osnovnih principih in ne na posebnostih posamezne uporabljene metode. Predvsem pa se moramo zavedati, da se večina razvojnih metod sorazmerno hitro spreminja. Zelo verjetno je zato tudi spoznanje, da čez tri leta nobena od uporabljenih metod ne bo več takšna, kot je danes.

Literatura:

- Barker, R., 1990a, *CASE*Method, Entity Relationship Modelling*, Addison-Wesley.
- Barker, R., 1990b, *CASE*Method, Tasks and Deliverables (2. Edition)*, Addison-Wesley.
- Barker, R., Longman, C., 1992, *CASE*Method, Function and Process Modelling*, Addison-Wesley.
- Booch, G., 1991, *Object-Oriented Design with Applications*, Addison-Wesley.
- Coad, P., Yourdon, E., 1991a, *Object-Oriented Analysis (2. Edition)*, Yourdon Press Computing Series, Prentice Hall, Inc.
- Coad, P., Yourdon, E., 1991b, *Object-Oriented Design*, Yourdon Press Computing Series, Prentice Hall, Inc.
- DeMarco, T., 1978, *Structured Analysis and System Specification*, Yourdon Press Computing Series, Prentice Hall, Inc.
- Jacobson, I., 1992, *Object-Oriented Software Engineering*, Addison-Wesley.
- Martin, J., Odell, I.J., 1993, *Principles of Object-Oriented Analysis and Design*, Prentice Hall, Inc., PTR Professional Technical Reference.
- Rumbaugh, J. et al., 1991, *Object-Oriented Modeling and Design*, Prentice Hall, Inc.
- Shlaer, S., Mellor, J.S., 1988, *Object-Oriented Systems Analysis: Modelling the World in Data*, Yourdon Press Computing Series, Prentice Hall, Inc.
- Shlaer, S., Mellor, J.S., 1992, *Object Lifecycles: Modelling the World in States*, Yourdon Press Computing Series, Prentice Hall, Inc.
- Yourdon, E.N., 1988, *Managing the System Life Cycle, (2. Edition)*, Yourdon Press Computing Series, Prentice Hall, Inc.
- Yourdon, E.N., 1989, *Modern Structured Analysis*, Yourdon Press Computing Series, Prentice Hall, Inc.
- Yourdon, E.N., 1992, *Decline&Fall of the American Programmer*, Yourdon Press Computing Series, Prentice Hall, Inc.

Recenzija: Tomaž Banovec
Jože Senegačnik (v delu)