

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**OBJEKTNO ORIENTIRANO MODELIRANJE
INFORMACIJSKEGA SISTEMA OPERATIVNEGA PROCESA
V CENTRU VODENJA ELEKTRODISTRIBUCIJE**

Ljubljana, junij 2005

Tomaž Slokar

IZJAVA

Študent Tomaž Slokar izjavljam, da sem avtor tega magistrskega dela, ki sem ga napisal pod mentorstvom prof.dr. Taliba Damija in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, 3. junija 2005

Podpis: _____

KAZALO VSEBINE:

1	UVOD	1
1.1	Namen magistrskega dela	2
1.2	Cilj magistrskega dela	3
1.3	Metode dela	3
1.4	Struktura poglavij	3
2	PREDSTAVITEV POSLOVNEGA OKOLJA	4
2.1	Deregulacija in odpiranje elektroenergetskega trga	4
2.1.1	Odpiranje trga z električno energijo v EU	4
2.1.2	Odpiranje trga z električno energijo v Sloveniji	5
2.2	Predstavitev podjetja elektrodistribucije	7
2.2.1	Osnovni podatki	7
2.2.2	Organizacijska struktura	8
2.2.3	Stanje in cilji podjetja na področju tržnih in reguliranih dejavnosti	9
2.3	Predstavitev dejavnosti Upravljanja distribucijskega omrežja	10
2.4	Center vodenja kot poslovno-procesno informacijsko središče	11
3	KLASIČNI OBSTOJEČI PROCESNI INFORMACIJSKI SISTEM CENTRA VODENJA DEES	12
3.1	Sistemi vodenja realnega časa	12
3.2	Opis obstoječe konfiguracije DCV Elektro Primorska	15
3.2.1	SCADA procesni računalniški sistem	16
3.2.1.1	Strojna oprema v DCV	16
3.2.1.2	Programska oprema v DCV	17
3.3	Predstavitev in analiza operativno - delovnih procesov DCV	18
3.3.1	Splošna predstavitev področij UDO	19
3.3.2	Analiza delovnih procesov UDO	20
3.4	Modeliranje procesov in procesnih podatkov DEES	22
3.4.1	Upravljanje kompleksnosti procesov in podatkov	23
3.4.2	Funkcijski pristop	24

3.4.2.1	Funkcijsko modeliranje procesnih podatkov	24
3.4.2.2	Proceduralno programiranje.....	27
3.4.2.3	Pomanjkljivosti funkcijskega pristopa.....	31
4	TEMELJNI KONCEPTI OBJEKTNEGA RAZVOJA INFORMACIJSKIH SISTEMOV.....	32
4.1	Splošno o informatizaciji poslovnih procesov	32
4.1.1	Prenova poslovnih procesov.....	32
4.1.2	Pristopi k zasnovi in razvoju informacijskih sistemov	35
4.1.3	Metodologije razvoja informacijskih sistemov.....	37
4.1.3.1	EPT metodologija	38
4.1.3.2	Metodologija življenjskega cikla sistema.....	41
4.2	Objektni pristopi k razvoju informacijskih sistemov.....	43
4.2.1	Osnovni pojmi objektno orientiranih pristopov	44
4.2.2	Poenoteni objektno orientirani modelirni jezik UML.....	47
4.2.2.1	Cilji in namen aplikacije UML	49
4.2.2.2	Predstavitev jezika UML.....	50
4.2.2.3	Gradniki UML.....	51
4.2.3	Poenoteni proces razvoja informacijskega sistema.....	53
4.2.3.1	Metodologija ROPES	59
4.2.3.2	Metodologija COMET	61
5	INFORMACIJSKI MODEL OPERATIVNEGA PROCESA V CENTRU VODENJA DEES	64
5.1	Nekaj o inženiringu zahtev za informacijski model sistema.....	64
5.2	Objektno modeliranje procesnih podatkov.....	67
5.2.1	Aparaturno-orientirano modeliranje	68
5.2.1.1	Strukturna sistemizacija signalov	68
5.2.1.2	Tehnološka dekompozicija EE postroja	71
5.2.1.3	Definiranje funkcij in lokacij uporabe signalov.....	75
5.2.2	Preslikava aparaturnega modeliranja v objektno paradigmo.....	75
5.2.2.1	Analogija med aparaturnim in objektnim modelom EEP	76
5.2.2.2	Objektno-orientirana strategija modeliranja	78

5.3	Objektno modeliranje delovnih procesov za podporo poslovanju	81
5.3.1	Problemski dokument	82
5.3.2	Objektno UML modeliranje procesov	83
5.3.2.1	Diagrami primerov uporabe	83
5.3.2.2	Diagrami statične strukture sistema.....	90
5.3.2.3	Diagrami dinamičnega modela sistema.....	96
5.3.2.4	Diagrami implementacije.....	99
6	SKLEP	100
7	LITERATURA IN VIRI.....	103
	PRILOGA.....	1
A.	Funkcionalna dekompozicija delovnih procesov	2
B.	Začetni diagram razredov.....	6
C.	Končni diagram razredov.....	9
D.	Diagrama stanj	12
E.	Zaporednostna diagrama	14
F.	Diagrama aktivnosti	15
G.	Seznam uporabljenih kratic	17

SEZNAM SLIK:

Slika 1: Odpiranje trga in konkurenca.....	5
Slika 2: Organizacijska struktura in udeleženci odprtega trga električne energije	6
Slika 3: Organigram podjetja	8
Slika 4: DCV informacijsko središče	12
Slika 5: Sistem realnega časa	13
Slika 6: Strogi in mehki realni čas.....	14
Slika 7: Funkcijsko orientirani model procesnih podatkov EEP	24
Slika 8: Funkcijska predstavitev signala kot vektorja.....	25
Slika 9: Proceduralno programiranje	28
Slika 10: Primer globalnega dostopa do skladovnega polja.....	29
Slika 11: Primer skladovnega objekta – polja	30
Slika 12: Primer globalnega dostopa do skladovnega seznama.....	30
Slika 13: Primer skladovnega objekta – povezanega seznama.....	31
Slika 14: IT sistem sodobnega elektro podjetja	39
Slika 15: Model življenjskega cikla sistema	41
Slika 16: Razvoj upravljanja podatkov	43
Slika 17: Arhitektura: program – baza podatkov.....	44
Slika 18: Predstavitev koncepta objekta.....	45
Slika 19: Razvoj UML	48
Slika 20: Proces uveljavljanja UML.....	49
Slika 21: Razčlenitev poenotenega procesa	55
Slika 22: ROPES razvojni proces	60
Slika 23: ROPES delovni izdelki iteracij	61
Slika 24: COMET življenjski cikel razvoja.....	62
Slika 25: Umestitev inženiringa zahtev v razvoj informacijskega sistema.....	64
Slika 26: Specifikacija zahtev s primerom uporabe	66
Slika 27: Specifikacija zahtev z zaporednostnim diagramom.....	66
Slika 28: Komponente objektnega modeliranja	67
Slika 29: Modeliranje tipa naprav.....	69
Slika 30: Aparaturna predstavitev signala	71
Slika 31: Primer dela EE postrojenja	72
Slika 32: Dekompozicija sistema	73
Slika 33: Procedura generiranja signalov	74
Slika 34: Diagram prehoda stanj za objekt <i>Izvodni ločilnik</i>	77
Slika 35: Koraki objektna strategije modeliranja EEP	78
Slika 36: Primer strukture G – S in W – P	78
Slika 37: Primer razrednega diagrama EEP	79
Slika 38: Podsystem Izvodno polje	80

Slika 39: Prikaz povezav podsistema EEP	81
Slika 40: Akterji v obravnavanem sistemu	85
Slika 41: Model sistema s temeljnimi primeri uporabe	86
Slika 42: Primer uporabe <i>Prijava okvare</i>	87
Slika 43: Primer uporabe <i>Lokalizacija okvare</i>	88
Slika 44: Primer uporabe <i>Odprava okvare</i>	89
Slika 45: Primer uporabe <i>Obveščanje in poročila</i>	90
Slika 46: Kandidati za razrede	91
Slika 47: Začetni diagram razredov	94
Slika 48: Končni diagram razredov	95
Slika 49: Zaporednostni diagram <i>Prijava okvare</i>	97
Slika 50: Diagram aktivnosti za primer <i>Prijava okvare</i>	98
Slika 51: Diagram komponent obravnavanega podsistema	99
Slika 52: Diagram porazdelitve obravnavanega podsistema	99

SEZNAM TABEL:

Tabela 1: Delež dejavnosti v poslovnih prihodkih in poslovnem izidu iz poslovanja v letu 2002	9
Tabela 2: Funkcijsko-tehnološki moduli	25
Tabela 3: Funkcijsko-opisani procesni signali	26
Tabela 4: Primer tipov naprav v EE postroju	69
Tabela 5: Nabor signalov za tipe naprave	70
Tabela 6: Asociacija naprav s tehnološkimi moduli	73
Tabela 7: Primer izpisa generiranih signalov	74
Tabela 8: Signali in lokacije njihove uporabe	75
Tabela 9: Preslikava aparaturnega v objektni koncept modeliranja	77

1 UVOD

V razvitem svetu splošno sprejeta globalizacijska paradigma prinaša v naše življenjsko okolje družbeno-ekonomske spremembe, katerih razsežnosti in vpliva na prihodnje generacije si danes še ni mogoče popolnoma zamišljati. Naj bodo ti vplivi pozitivni ali negativni, dejstvo je to, da tako epohalnim spremembam nobena generacija človeštva še ni bila priča. Če odmislimo vplive na človeka in njegovo duhovno ter duševno komponento, pomenijo za gospodarstvo in akterje v njem te spremembe priložnost za uveljavitev in rast, obenem pa tudi nevarnost za stagnacijo in propad. Prelomno točko med obema možnostima predstavlja sposobnost podjetij za hitro prilagajanje novim razmeram in preseganje tradicionalnih načinov poslovanja ter eksploatacija tehnološkega razvoja, ki mu skorajda nismo kos slediti. Podjetja tako za svojo rast in razvoj ne morejo ignorirati novih pogojev poslovanja in se izogniti notranjim kulturnim, organizacijskim, poslovnim in tehnološkim spremembam.

V bodoči razširjeni Evropski uniji se bomo morali oziroma se tudi slovenski prebivalci in podjetja že privajamo na skupen notranji trg in pravila delovanja na njem. Prost pretok blaga, oseb, kapitala in storitev so temelj globalnega povezovanja, saj krepijo povezanost in vpetost držav v skupni evropski prostor ter prinašajo tudi številne prednosti za širitev poslovanja podjetij.

Odpiranju in sproščanju notranjih trgov, ki jih kot trge, na katerih vlada konkurenca, že dolgo poznamo, je sledilo tudi odpiranje notranjih reguliranih trgov z energenti, najprej trga z električno energijo. Procesu liberalizacije trga z električno energijo v ZDA in nekaterih južnoameriških državah so, zaradi ohranjanja konkurenčnosti svojih podjetij, sledile tudi države Evropske unije, dogajanju na tem področju pa aktivno sledi tudi Slovenija.

Slovenski Energetski zakon deregulira obstoječe stanje elektroenergetskega sistema Slovenije in postavlja izhodišča za drugačno organizacijsko strukturo elektrogospodarstva. Zakon omogoča vzpostavitev konkurence najprej na notranjem trgu, in sicer na področjih proizvodnje in dobave električne energije, kasneje pa še odprtje trga navzven, ter regulira dejavnosti prenosa in distribucije električne energije kot naravnih monopolov (Energetski zakon, 1999).

Slovenska podjetja za distribucijo električne energije doslej niso bila podvržena tržnim zakonitostim in niso bila prisiljena delovati v smeri strateškega načrtovanja. Poleg neobstoječega trga na področju distribucije električne energije je k omejenemu angažiranju podjetij prispevala tudi lastniška struktura podjetij in sama organizacija elektrodistribucije. Nova energetska zakonodaja s svojimi podzakonskimi akti predpisuje poleg delitve na tržne in regulirane dejavnosti tudi medsebojno ločevanje reguliranih dejavnosti, tako organizacijsko kot tudi finančno-računovodsko. Kljub reguliranosti področij in statusa naravnih monopolov so se tudi regulirane dejavnosti v določeni meri prisiljene tržno obnašati in izkoriščati svoje neizrabljene potencialne, predvsem v luči bodoče delne privatizacije elektrodistribucijskih podjetij. Od njih se poleg izvajanja svoje osnovne dejavnosti na področju reguliranih gospodarskih javnih služb pričakuje predvsem racionalizacija ter optimizacija poslovanja in stroškov, povečanje učinkovitosti in produktivnosti ter trženje in širjenje svojih dopolnilnih dejavnosti in storitev.

Gospodarska javna služba (GJS) Upravljanja distribucijskega omrežja (UDO) je kot regulirana dejavnost odgovorna za vodenje in obratovanje omrežja, omogočanje dostopa do omrežja ter obdelavo merilnih podatkov odjemalcev (Uredba o načinu izvajanja javnih

gospodarskih služb s področja distribucije električne energije, 2000). V okviru svojih odgovornosti opravlja naslednje naloge (Upravljanje distribucijskega omrežja, 2003):

- spremljanje in nadzor nad pretoki energije,
- usklajevanje vzdrževalnih in drugih posegov v omrežju,
- nadzor in vzdrževanje naprav za daljinsko vodenje, zaščito in telekomunikacije,
- izdaja soglasij za priključitev,
- priprava pogodb za dostop do distribucijskega omrežja za upravičene odjemalce in
- druge naloge, ki izhajajo iz Energetskega zakona in podzakonskih aktov.

V okviru organizacijske strukture GJS UDO predstavlja Distribucijski center vodenja (DCV) poglavitno informacijsko središče in stičišče za večino nalog, ki jih pokriva GJS UDO. V splošnem sodijo sodobni centri vodenja distribucijskega elektroenergetskega sistema (DEES) med procesno in tehnološko najzahtevnejše sisteme vodenja, tako v primerjavi z drugimi nadzornimi centri znotraj elektroenergetskega sistema kot tudi širše. To dejstvo potrjuje število procesnih podatkov, ki se v sistemu obdelujejo, in je posledica vpetosti med vse akterje elektroenergetskega sistema in trga. Prav ogromno število operativnih in sistemskih podatkov, ki se generirajo v elektroenergetskem sistemu, ter število delovnih procesov, ki jih morajo obvladovati operaterji v centru vodenja, botruje zasnovi, razvoju in implementaciji informacijskega sistema, ki bo operaterjem omogočil učinkovito delo, podjetjem pa boljše poslovanje.

1.1 Namen magistrskega dela

Namen magistrskega dela je prikazati drugačen pogled na celoten DEES in doprinesiti k spoznanju, da je potrebno pri izgradnji sodobnega centra vodenja upoštevati v tem delu predlagani koncept. Temeljna hipoteza magistrskega dela je torej, da predlagani koncept pomeni bistveno povečano učinkovitost modeliranja procesa in podatkov, učinkovitejše izvajanje nadgradenj in vzdrževanja sistema ter na koncu, vendar ne nazadnje, tudi učinkovitejše vodenje DEES. Miselni preskok pri prehodu iz klasičnega pogleda na objektno orientirani pogled DEES bo za konzervativno okolje elektroenergetike sicer težak, zato pa je potrebno problematiko predstaviti na način, kjer bodo jasno vidne prednosti predlaganega pristopa pri uporabi na področju operativnega vodenja DEES ter tudi pri izobraževanju in logičnem razumevanju delovanja DEES (Šimunić, 1997, str. 207).

Postavljen informacijski sistem je dandanes potreben pogoj za poslovanje vsakega podjetja, ni pa še zadosten. Potreben je zato, ker preko informacijskega sistema kakršnakoli organizacija lahko krepko poveča učinkovitost, produktivnost in dobičkonosnost. Ni pa zadosten zato, ker vsako podjetje tvori predvsem človeški dejavniki, ki jim mora biti omogočeno učinkovito delo in morajo tudi sami učinkovito delati. Samo izpolnjena oba pogoja doprineseta k zastavljenim ciljem (Debs, 1997, str.73). Temelj za tak informacijski sistem so informatizacija vseh delovnih procesov znotraj DCV, ki do tega trenutka niso bili ne informatizirani ne avtomatizirani, ter izgradnja takega modela procesnih in sistemskih podatkov, ki bo operaterjem v pomoč in podporo pri odločanju. Vodenje DEES namreč obsega nadzor, krmiljenje in odločanje, prav slednje pa je ključno za doseganje zastavljene učinkovitosti (Gubina, 1997, str. 22-25).

Danes velja za privzeto in tudi na mnogih področjih dokazano, da je za doseg omenjenih ciljev možno izvesti informatizacijo procesov ter modeliranje podatkov z uporabo objektne paradigme. Objektna paradigma namreč združuje tisto, kar je bilo do sedaj ločeno, in sicer modeliranje procesov in modeliranje podatkov (Grad, 1996, str. 115).

1.2 Cilj magistrskega dela

V luči zahtev energetske zakonodaje in glede na že omenjeno delno privatizacijo je znotraj podjetij elektrodistribucije prišlo do spoznanja, da je investicija v novo informacijsko tehnologijo ne samo možnost, ampak predvsem nuja, ki bo podjetjem sploh omogočila obstanek in preživetje. Elektrodistribucijsko podjetje bo izpostavljeno tržni konkurenci tudi na reguliranem področju, kjer bo učinkovito obvladovanje elektroenergetskega sistema in s tem večja kvaliteta dobavljene energije konkurenčna prednost.

V magistrskem delu želim predstaviti objektno-orientirani pristop k razvoju informacijskega sistema in podati prednosti, ki jih tak pristop prinaša v primerjavi z dosedanjim načinom modeliranja. Osnova za to bo objektno modeliranje procesov in podatkov v okviru izbranih faz procesa razvoja informacijskega sistema. Osredotočil se bom na analizo zahtev in analizo obravnavanega sistema, ki sta tudi osnova za kasnejše načrtovanje in implementacijo. Ugotovitve in smernice želim prenesti na konkretno aplikacijo centra vodenja DEES v podjetju elektrodistribucije. Želim podati tudi konkretne dobrobiti, ki jih bo postavitev takega informacijskega sistema prinesla poslovanju družbe.

1.3 Metode dela

Metoda dela, ki jo bom uporabil pri izdelavi magistrskega dela, bo temeljila na povezavi teoretičnih zaključkov izbrane strokovne literature in virov s področja objektno-orientiranih informacijskih sistemov ter analitičnih in praktičnih izkušenj tako posameznih avtorjev kot tudi izdelovalca magistrskega dela na področju operativnega vodenja DEES, v sklop katerega sodi tudi obvladovanje obstoječega procesnega informacijskega sistema. Spoznanja in pozitivne izkušnje, ki so se pokazala pri aplikacijah objektno-orientirane paradigme na področju računalniškega programiranja in na področju telekomunikacijskih sistemov, želim prenesti tudi na področje obvladovanja elektroenergetskih sistemov.

1.4 Struktura poglavij

V prvem delu bo predstavljeno okolje povezano z obravnavano problematiko. V drugem delu bo predstavljen obstoječ procesni informacijski sistem za potrebe vodenja DEES in klasični način modeliranja podatkov, v tretjem delu pa bo za potrebe predstavitve objektno-orientiranega procesa prenove in izgradnje informacijskega sistema prikazana uporaba metodologij razvoja programske opreme ter UML notacija. V četrtem delu bo predstavljena zasnova enega dela poslovno-procesnega informacijskega sistema novega DCV na osnovi nove modelirne zasnove. Podane bodo prednosti novega objektno-orientiranega pristopa za potrebe modeliranja procesov in podatkov ter za potrebe zasnove bodočega informacijskega

sistema. V zaključnem poglavju bom predstavil sklepe, do katerih me bodo pripeljali rezultati analize obravnavanega področja, ter konkretne dobrobiti predlaganega pristopa.

2 PREDSTAVITEV POSLOVNEGA OKOLJA

2.1 Deregulacija in odpiranje elektroenergetskega trga

Na začetek procesa liberalizacije, t.j. deregulacije in s tem odpiranja elektroenergetskega trga, je v največji meri vplivala želja po povečanju konkurenčnosti na domačem in svetovnem trgu električne energije. Širši pojem liberalizacije namreč pomeni sprostitev oziroma odstranitev regulatornih omejitev v poslovanju, kar posledično na trg prinese prosto konkurenco. Le-ta predstavlja enega izmed temeljev ureditve Evropske unije in njenega gospodarskega razvoja. Liberalizacija in v ožjem pomenu deregulacija elektroenergetskega sektorja je torej svetovni proces, ki ima v Evropi tri pomembne vzroke, in sicer političnega, ekonomskega in pravnega (Klom, 1997, str. 86-90).

V političnem kontekstu Evropa tako ali drugače tekmuje z ostalim svetom. Z ekonomskega vidika evropska podjetja tekmujejo z drugimi svetovnimi proizvajalci, ki imajo drugačna ekonomska okolja, predvsem pa nižje cene električne energije, kar jim posledično prinaša nižjo ceno končnih proizvodov in s tem večjo konkurenčnost. Povečanje konkurenčnosti industrijskih podjetij je zato eden izmed glavnih ciljev razvoja skupne evropske energetske politike. Tretji razlog za deregulacijo trga v Evropi je pravne narave, saj Evropski sporazum določa notranji evropski trg kot trg brez notranjih meja, kjer je zagotovljen prost pretok blaga, oseb, storitev in kapitala. Ker je električna energija v luči že povedanega pojmovana kot blago, njena prodaja pa kot storitev, torej ne more biti izvzeta iz konteksta Evropskega sporazuma.

2.1.1 Odpiranje trga z električno energijo v EU

Proces odpiranja trga z električno energijo je v posameznih državah EU potekal različno, s tem da je cilj liberalizacije in deregulacije ostal nespremenjen in enoten. V nekaterih državah je proces odpiranja trga vključeval vzpostavitev konkurence v posamezni državi, v drugih je proces odpiranja trga potekal vzporedno s procesom internacionalizacije (Hrovatin, 2003, str.3).

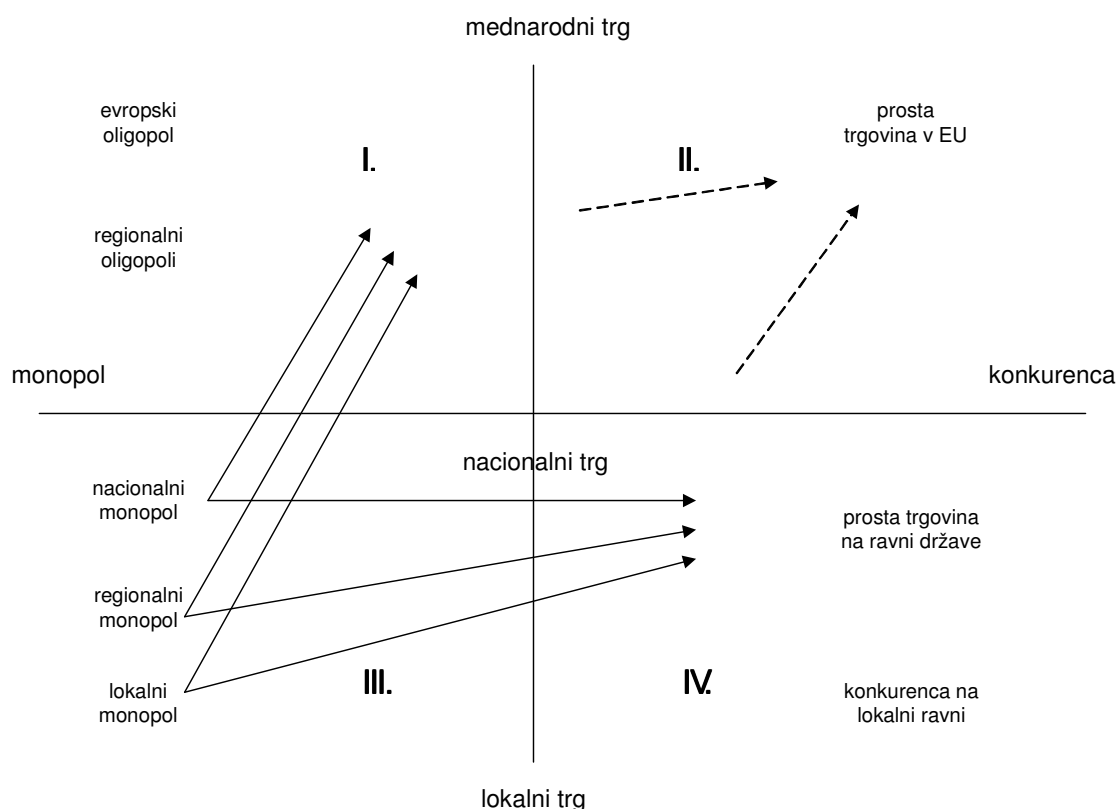
Temelj za deregulacijo trga z električno energijo v Evropski uniji je predstavljala Direktiva EU o skupnem notranjem trgu električne energije (Directive 96/92/EC, 1996), ki je stopila v veljavo 19.02.1999. Direktiva je bila z izdajo nove Direktive (Directive 2003/54/EC, 2003) z dne 26.06.2003 sicer razveljavljena, vendar pa v vsebinskem smislu deregulacije trga ne spreminja, saj določa le pravne in organizacijske okvire delovanja akterjev na notranjem trgu EU. Direktivi sta odraz pogajanj in pridobljenih izkušenj in kot taki predstavljata kompromis med članicami EU, predvsem med Veliko Britanijo in Nemčijo na eni strani ter Francijo na drugi.

Evropska unija je z implementacijo politike deregulacije in liberalizacije na področju električne energije pričela v februarju 1999 in s tem ustvarila osnovo za novo strateško

konfiguracijo evropskih energetskih podjetij. Tradicionalne omejitve, s katerimi so bila soočena energetska podjetja, t.j. sektorske in geografske omejitve ter omejitve glede organizacijske oblike, so se močno zrahlijale ter so bile ponekod deloma tudi odpravljene. Tako npr. regulatorni organi v večini držav EU dovoljujejo povezovanje podjetij, ki delujejo na področju elektroenergetike, s podjetji na področju oskrbe s plinom in nafto, s telekomunikacijskimi podjetji ter podjetji na področju druge infrastrukture kot npr. vodnem, transportnem ipd. Poleg tega je odprava nacionalnih ovir za trgovanje podjetjem prinesla možnost vstopa tudi na druge trge zunaj njihovih tradicionalnih preskrbovalnih območij ter preko nacionalnih meja.

V luči postopnega odpiranja in internacionalizacije trga z električno energijo se evropska podjetja pomikajo v smeri nadnacionalnih podjetij preko mehanizmov povezav, prevzemov in združitvev (Hrovatin, 2003, str.2). Prav to utegne na dolgi rok pripeljati do problema tržne koncentracije na evropski ravni in do zmanjšanja konkurenčnih pritiskov na podjetja, kar pa je bil temeljni namen deregulacije sektorja. Sam po sebi se tako vsiljuje zaključek, da je trenutna konkurenca na evropskem trgu z električno energijo dejansko le vmesna faza na poti med stanjem nacionalnih monopolov in evropskih oligopolov (slika 1).

Slika 1: Odpiranje trga in konkurenca



Vir: Midttun, 2001, str. 17.

2.1.2 Odpiranje trga z električno energijo v Sloveniji

Velikost in način odpiranja nacionalnega trga je v veliki meri pogojena z velikostjo elektroenergetskega sektorja v posamezni državi. Obstajajo tradicionalne razlike med

Glavne zakonitosti delovanja trga z energijo so predpisane z Energetskim zakonom (Uradni list RS, št. 79/99) in njegovimi podzakonskimi akti. Energetski zakon (EZ) določa načela energetske politike, pravila za delovanje trga z energijo, načine in oblike izvajanja gospodarskih javnih služb na energetskem področju, načela zanesljive oskrbe in učinkovite rabe energije ter pogoje za obratovanje energetskih sistemov, pogoje za opravljanje energetske dejavnosti, pogoje izdajanja licenc in dovoljenj ter organe, ki opravljajo upravne naloge po Energetskem zakonu. Na področju deregulacije elektroenergetskega sektorja zakon navaja tudi udeležence trga električne energije, organizatorje trga, nadzorne organe in upravljalce omrežij, po katerih se tržno blago - električna energija pretaka.

Po vsebinski plati EZ omogoča prehod iz s strani države tržno-reguliranega sistema na prosto-tržni sistem na področju nakupa in prodaje električne energije, kar pomeni, da ima potrošnik oziroma odjemalec električne energije možnost proste izbire dobavitelja, hkrati pa se tudi cene prosto oblikujejo na organiziranem trgu z električno energijo.

6

predmetno magistrsko delo, ki sicer zaobjema informacijski sistem centra vodenja elektrodistribucije v smislu tehničnega obvladovanja elektroenergetskega sistema, navezuje tudi na področje obvladovanja širšega informacijskega sistema v razmerah odprtega elektroenergetskega trga, o čemer pa bo govora v nadaljevanju.

2.2 Predstavitev podjetja elektrodistribucije

Elektrogospodarstvo, z njim pa tudi podjetja elektrodistribucije, je skozi pretekla desetletja doživljalo številne reorganizacije, integracije in razdružitve. Vse te spremembe so prvenstveno odsevale le bolj ali manj trenutna politična razmerja, ki so nedvomno bila tudi, na srečo ali na žalost, del dejavnikov odločanja.

Popolnoma druge razmere so nastopile z nastopom tesnejših integracijskih procesov v Evropi in splošno privzete globalizacijske paradigme. Direktive Evropske unije in na njih temelječa domača zakonodaja zahtevajo reorganizacijo celotnega energetskega sektorja, tokrat poleg v formalno-pravnem tudi in predvsem v vsebinskem pomenu. Pri tem velja poudariti, da so pri vzpostavljanju in uspešni rabi nove organizacijske strukture najpomembnejši vplivni dejavniki strategija poslovanja, vrsta uporabljene tehnologije, kadrovska struktura, stopnja diverzifikacije, velikost, tradicija, vodstvo in drugi (Možina, 1994, str.377). Upoštevajoč navedeno je v podjetjih elektrodistribucije prišlo do spoznanja, da se morajo večinoma toge in vodstveno centralizirane organizacijske strukture prilagoditi novemu poslovnemu okolju in pogojem poslovanja.

2.2.1 Osnovni podatki

Pri opisu se bom osredotočil na elektrodistribucijsko podjetje, kjer sem trenutno zaposlen, t.j. Elektro Primorska, javno podjetje za distribucijo električne energije, d.d.

Sedanja organizacijska oblika podjetja je delniška družba, ki opravlja javne gospodarske službe s področja distribucije električne energije in je bila ustanovljena 18. februarja 1998. Podjetje Elektro Primorska, d.d. je kot javno podjetje za distribucijo električne energije del slovenskega elektroenergetskega sistema in v svojem preskrbovalnem območju pokriva JZ, Z in SZ del Slovenije. Velikost celotnega preskrbovalnega območja znaša 4.335 km². Po velikosti je četrto elektrodistribucijsko podjetje v Sloveniji. Leta 2002 je podjetje distribuiralo 15 % električne energije v Sloveniji (Statistični letopis energetskega gospodarstva RS 2002, 2003, str. 174).

V letu 2002 je podjetje oskrbovalo 117.608 odjemalcev, ki so skupno prevzeli 1.254.344 MWh električne energije. Delež 708 upravičenih odjemalcev je znašal 700.859.234 kWh, ki so energijo odjemali iz 1.157 prevzemno-odjemnih mest.

Podjetje posluje kot delniška družba. Večinski lastnik delnic je Republika Slovenija z 79,5% deležem. Poleg RS so lastniki delnic še:

- | | |
|--|-----------|
| • Kapitalska družba, d.d. | 0,7488 % |
| • Kapitalska družba, d.d. - PPS | 0,7648 % |
| • Pooblašcene investicijske družbe (PID) | 18,4958 % |
| • Zaposleni, bivši zaposleni in upokojenci | 0,4906 %. |

2.2.2 Organizacijska struktura

Na podlagi Energetskega zakona in Uredbe o načinu izvajanja gospodarskih javnih služb s področja distribucije električne energije je moralo podjetje ločiti svoje energetske dejavnosti, zanje pridobiti licence in spremeniti organizacijsko strukturo v skladu z novimi zahtevami. Tako so v okviru podjetja od 1.1.2001 organizacijsko in informacijsko razdeljene naslednje dejavnosti:

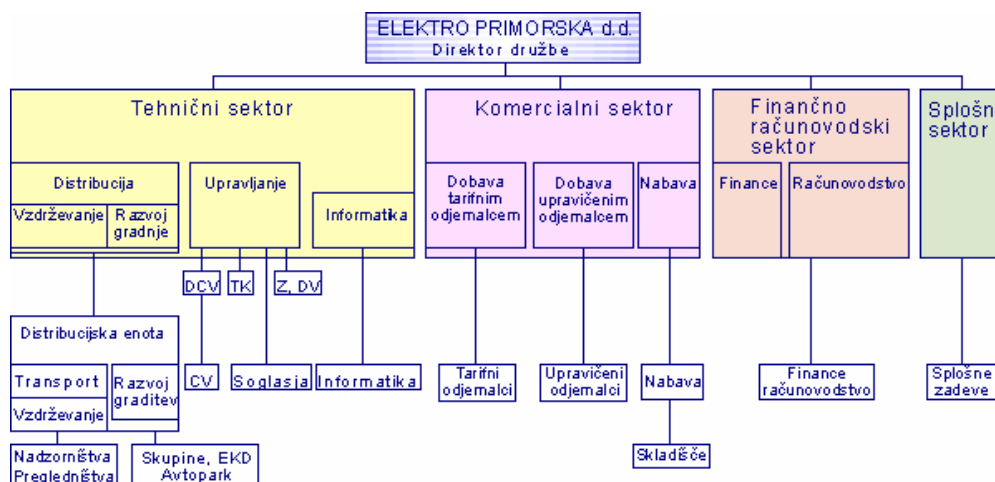
1. Gospodarske javne službe:
 - dejavnost distribucije električne energije
 - dejavnost upravljanja distribucijskega omrežja
 - dejavnost dobave električne energije tarifnim odjemalcem
2. Energetske tržne dejavnosti:
 - dejavnost dobave električne energije upravičenim odjemalcem
3. Neenergetska tržna dejavnost:
 - dejavnost storitev
4. Skupne štabno-strokovne službe.

Podjetje je znotraj organizirano v štiri sektorje: tehnični sektor, komercialni sektor, finančno-ekonomski sektor in splošni sektor.

Teritorialno opravlja podjetje na lokacijah Kopra, Nove Gorice, Sežane in Tolmina naslednje dejavnosti:

- distribucijo električne energije
- upravljanje distribucijskega omrežja
- prodajo električne energije tarifnim odjemalcem in
- tržno dejavnost.

Slika 3: Organigram podjetja



Vir: Predstavitev organiziranosti Elektro Primorska d.d., 2003.

2.2.3 Stanje in cilji podjetja na področju tržnih in reguliranih dejavnosti

Elektro Primorska, d.d. je organizirana kot enovito podjetje brez odvisnih oziroma pridruženih podjetij. Podjetje opravlja dve osnovni vrsti dejavnosti, in sicer gospodarske javne službe ter tržne dejavnosti. Pomen posameznih dejavnosti za podjetje prikazuje njihova udeležba v celotnih poslovnih prihodkih in v poslovnem izidu iz poslovanja, kar je prikazano v Tabeli 1.

Tabela 1: Delež dejavnosti v poslovnih prihodkih in poslovnem izidu iz poslovanja v letu 2002

(v 000 SIT)

	DISTRIBUCIJA		UPRAVLJANJE		DOBAVA TO		TRŽNE DEJAVNOSTI		SKUPAJ
Kategorija	Vrednost	Delež	Vrednost	Delež	Vrednost	Delež	Vrednost	Delež	Vrednost
Poslovni prihodki	5.697.561	25,5%	4.011.733	17,9%	4.462.795	19,9%	8.207.821	36,7%	22.379.910
Poslovni izid iz poslovanja	153.055		270.352		-1.250.499		951.829		124.737
ROA ¹ (%)	0,5		20,4		-94,4		41,1		
ROE ² (%)	0,5		41,2		-234,5		55,9		

1- ROA: donosnost sredstev (return on assets) = (čisti dobiček / sredstva)*100

2- ROE: donosnost kapitala (return on capital) = (čisti dobiček / kapital)*100

Vir: Letno poročilo družbe za leto 2002, 2003, str.77.

Največji delež poslovnih prihodkov v letu 2002 so družbi prispevale tržne dejavnosti. Kot glavno velja omeniti dobavo električne energije upravičenim odjemalcem (36,7%), sledijo ji dejavnosti distribucije (25,5%), dobave tarifnim odjemalcem (19,9%) in upravljanja distribucijskega omrežja (17,9%). Tržna dejavnost je izkazovala tudi največjo donosnost z visoko nadpovprečnim donosom na kapital (41,2%). Zaradi prenizko določenih cen za prodajo električne energije tarifnim odjemalcem s strani Vlade RS je veliko izgubo podjetju prinašala dejavnost dobave električne energije tarifnim odjemalcem. Omenjeno dejavnost je podjetje skladno z zakonodajo dolžno izvajati kot gospodarsko javno službo. Distribucija, kot druga dejavnost po višini prihodkov, ima sorazmerno majhen donos (0,5%). Pri tem velja opozoriti na obvezno delitev dejavnosti po Energetskem zakonu na dejavnost distribucije in upravljanja, s katero je posledično povezana tudi delitev prihodka in sredstev med obe dejavnosti. Ker sta omenjeni dejavnosti v neposredni povezavi, bi bilo mogoče spremljati tudi njuno donosnost celovito. V tem primeru bi skupna donosnost obeh dejavnosti znašala (ROA=1,3%, ROE=1,46%), kar je sicer nekoliko bolj zadovoljivo, vendar še vedno dokaj nizko.

Trenutno dejavnosti, ki jih podjetje opravlja, kažejo na ozko specializacijo na področjih distribucije in dobave električne energije ter upravljanja električnega omrežja. Takšna specializacija z relativno majhnim obsegom dejavnosti na enem geografskem področju prinaša podjetju številne slabosti:

- velika odvisnost prihodkov in poslovnega rezultata od ozkega obsega dejavnosti
- velika odvisnost uspešnosti dejavnosti od ustreznega delovanja regulatornih organov
- nezmožnost izkoriščanja ekonomij skupne dejavnosti
- nezmožnost izkoriščanja managerskih sposobnosti za širjenje v sorodne dejavnosti

- nefleksibilnost podjetja in togost organizacijske strukture
- nepodjetniška klima
- omejene možnosti pritegnitve strateških partnerjev in
- posledično omejene možnosti rasti in razvoja.

Iz navedenega lahko ugotovimo, da takšna organiziranost danes ni več primerljiva s sodobnimi podjetji v Evropi, ki so se v procesu prilagajanja novim razmeram in diverzifikacije preobrazila v »multi-utility« podjetja z razvejanimi dejavnostmi.

Elektro Primorska d.d. ima vizijo, da postane sodobno organizirano podjetje, ki združuje več energetske dejavnosti oziroma širi dejavnosti tudi na druga področja. Na področju tržnih dejavnosti mora podjetje utrditi položaj enega pomembnejših trgovsko-storitvenih podjetij v slovenskem prostoru in se razširiti na sosednja tržišča Evropske unije. Med prve korake na poti k temu sodi oblikovanje korporacijske strategije oziroma strategije na ravni podjetja, ki mora vključevati lasten razvoj, vzpostavljanje mednarodnih povezav, prilagodljivost in učinkovitost. Prav tako mora prispevati k temu, da bodo vse dejavnosti v portfelju podjetja dosegale večje donose kot bi jih v primeru, če takšna strategija ne bi obstajala. Primarna pot za izvajanje korporacijske strategije je diverzifikacija, ki od managementa zahteva izvajanje strategije »multi dejavnosti«.

Na področju regulirane dejavnosti mora podjetje predvsem dvigniti nivo notranjega poslovanja in poslovanja z odjemalci električne energije. Sem sodi prilagoditev standardom kakovosti, oblikovanje organizacijske strukture, ki bo omogočala tesnejšo povezavo in sodelovanje z odjemalci in ki bo izkoriščala dinamičnost zaposlenih. Cilj podjetja je dosegati zadovoljiv donos na vloženi kapital, ki bo omogočal zadosten razvoj in s tem zadoščanje naraščajočim potrebam energetskega sektorja. V luči tehnične občutljivosti elektrodistribucijskega sistema mora podjetje zagotoviti dovolj sredstev za vzdrževanje sistema skozi ceno za uporabo omrežja, predvsem z argumentiranim izkazovanjem potreb in racionalizacijo t.i. ostalih stroškov poslovanja. To zadnje mora predvsem biti izraženo skozi večjo učinkovitost managementa in zaposlenih.

2.3 Predstavitev dejavnosti Upravljanja distribucijskega omrežja

Področje dejavnosti Upravljanja distribucijskega omrežja podrobneje ureja Uredba o načinu izvajanja gospodarskih javnih služb s področja distribucije električne energije. Definira njeno organizacijsko zasnovo, pravice uveljavljanja dostopa do distribucijskega omrežja, pravice in pogoje priključevanja odjemalcev na distribucijsko omrežje, financiranje dejavnosti in drugo. Javna služba se dodeli s koncesijo s strani Vlade RS za določen čas in obseg: vodenje in obratovanje distribucijskega omrežja, zagotavljanje dostopa do omrežja upravičenim odjemalcem in proizvajalcem električne energije ter izvajanje sistemskih obratovalnih navodil. Možnost priključitve na distribucijsko omrežje ima vsak proizvajalec ali upravičeni odjemalec¹ električne energije, ki se glede na napetostni nivo svoje proizvodnje ali odjema lahko priključi. Tehnične in druge pogoje za priključitev določajo sistemska obratovalna navodila. Upravičeni odjemalci in proizvajalci električne energije imajo pravico dostopa do distribucijskega omrežja za izvršitev pogodb, evidentiranih na organiziranem trgu z električno

¹ odjemalec, ki ima pravico do svobodne izbire dobavitelja energije na trgu

energijo ali sklenjenih na tem trgu, razen če jim upravljavec omrežja (sistemski operater)² dostop zavrne. Upravljavec omrežja lahko zavrne dostop le zaradi tehničnih in obratovalnih omejitev, enkrat mesečno pa mora na svoje stroške objaviti podatke o zasedenosti distribucijskega omrežja po posameznih distribucijskih smereh ter hkrati varovati zaupne ali poslovno občutljive podatke. Dostop do omrežja lahko uveljavljajo tudi trgovci z električno energijo, in sicer s sklenitvijo pogodbe ali s trenutnim dostopom do omrežja v okviru izravnave odstopanj. Uporabnik omrežja je dolžan plačati pogodbeno zagotovljeni dostop do omrežja v celoti, ne glede na to, koliko električne energije je bilo v okviru tega dostopa ob posameznem času preneseno.

Kot upravjalci distribucijskega omrežja so s posebnim sklepom Vlade RS (koncesijo) imenovana obstoječa javna podjetja za distribucijo električne energije Elektro Ljubljana, d.d., Elektro Maribor, d.d., Elektro Celje, d.d., Elektro Gorenjska, d.d. in Elektro Primorska, d.d. Ustanovljena so za opravljanje distribucije električne energije na območjih, ki so določena v njihovih dosedanjih aktih in so imela v letu 1999 odvzem 8.245 GWh električne energije, kar predstavlja 66% celotno odvzete oziroma porabljene električne energije iz omrežja, poleg odvzema s strani neposrednih odjemalcev, izvoza in izgub električne energije (Annual Report ELES, 1999, str. 28).

2.4 Center vodenja kot poslovno-procesno informacijsko središče

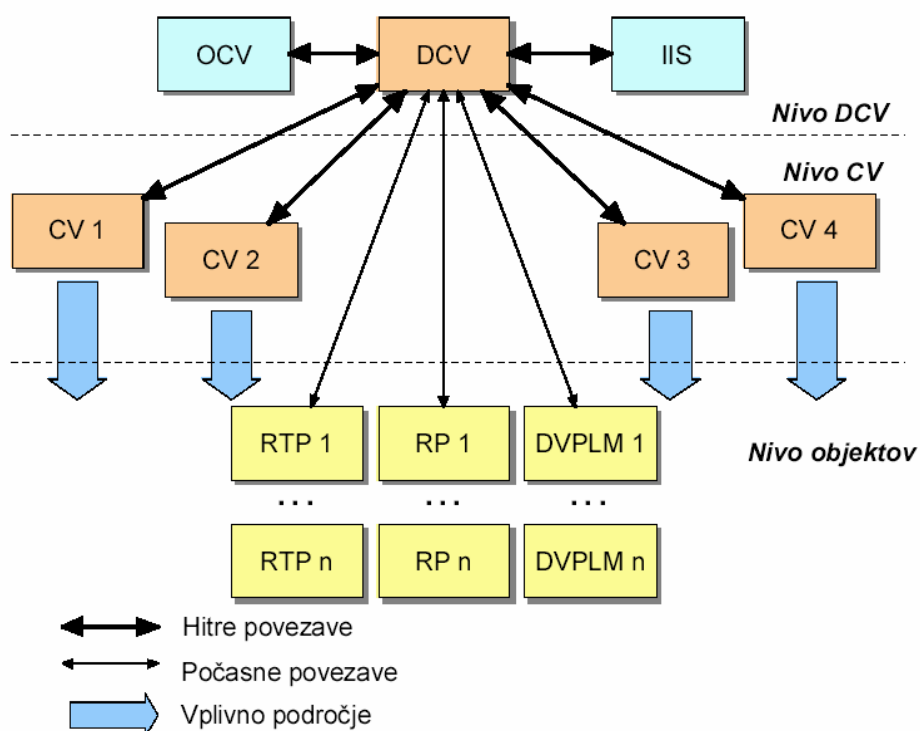
Distribucijski center vodenja (DCV) predstavlja poglavitno informacijsko središče za večino nalog, ki jih pokriva GJS UDO. Distribucijski centri vodenja preko svojih obstoječih nadzornih sistemov vodenja sprotnega (realnega) časa³ razpolagajo s procesnimi podatki o stanju distribucijskega omrežja, s podatki o stanju na napravah, ki niso vključene v daljinski nadzor, s tehničnimi podatki o elektroenergetskih napravah, s podatki o odjemalcih električne energije, s podatki, ki so zunanjega značaja (vremenski podatki, podatki o delu na drugih objektih v neposredni bližini energetskega objekta, podatki iz lokalnih OKC-jev ipd.). V okviru centra vodenja se izvajajo tudi analize omrežja, in sicer na realnih arhiviranih podatkih ali pa na simuliranih podatkih. Na ta način se sproti preverja možnosti in meje obratovanja omrežja. Rezultati tovrstnih izračunov so tako vhodni podatek za razvojne študije oziroma planiranje razvoja omrežja. Poleg obvladovanja procesnih podatkov igra DCV lahko tudi vlogo klicnega centra v smislu sprejemanja prijav okvar na elektroenergetskih napravah in posredovanja informacij odjemalcem ter koordinatorja vzdrževalnih del na EE napravah.

V nadaljevanju bom podrobno predstavil obstoječo informacijsko tehnologijo centrov vodenja v podjetjih elektrodistribucije.

² pravna oseba, ki vodi omrežje na določenem območju

³ real-time sistemi

Slika 4: DCV informacijsko središče



Vir: Idejni projekt DCV Elektro Primorska, 1999, str.37.

3 KLASIČNI OBSTOJEČI PROCESNI INFORMACIJSKI SISTEM CENTRA VODENJA DEES

3.1 Sistemi vodenja realnega časa⁷

Že sama narava operaterjevega dela v centru vodenja distribucijskega elektroenergetskega sistema v osnovi pomeni odzivanje na zunanje dogajanje, ki mora biti takojšnje in ažurno. Dogodek v omrežju, zunanji telefonski klic ali obvestilo dežurnih terenskih ekip je za operaterja v DCV prožilo, ki zahteva takojšnje ukrepanje. Drugi, aparaturni del takojšnjega odzivanja, pa opravlja strojna in programska oprema, katere končna destinacija njene obdelave je operater v DCV. Takojšnjemu in sprotnemu odzivanju strojne in programske opreme ter obdelavi vhodnih podatkov pravimo delovanje v realnem času.

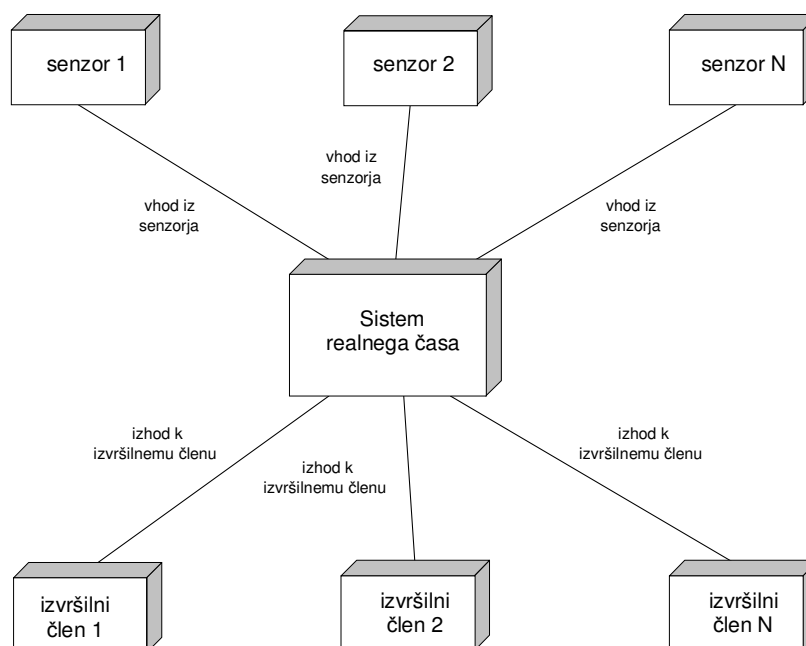
Pod pojmom sistema realnega časa običajno razumemo samo sistemsko aplikacijo realnega časa, operacijski sistem, ki se izvaja v realnem času, ter največkrat tudi posebno sistemsko programsko opremo, kot je npr. komunikacijska oziroma gonilniška programska oprema (Gomaa, 2000, str. 9). Za razumevanje samega pojma realnega časa podajamo dve definiciji realnega časa:

⁷ real time systems

- *Sistem deluje v realnem času, če so programi sposobni sproti obdelovati podatke, ki prihajajo od zunaj, in so rezultati teh obdelav na voljo v vnaprej določenih časovnih intervalih. Časi prihodov podatkov so pri tem lahko razporejeni naključno ali pa vnaprej določeni, kar je odvisno od aplikacije. (DIN 44300, 1988).*
- *Obdelava v realnem času je tista, katere pravilnost ni odvisna le od logičnih rezultatov, temveč tudi od časa, ko so bili ti rezultati dobljeni (Colnarič, 2002, str. 1)*

Računalniški sistemi, delujoči v realnem času (v grobem jih lahko poimenujemo krmilni sistemi), izvršujejo programe, ki so povezani s procesi iz okolja (periferni krmiljeni sistemi). Obdelava teh programov mora biti sinhronizirana z iz periferije izhajajočimi zunanji dogodki. Sistemi v realnem času so torej del širšega okolja oziroma so vanj vgrajeni, zato jim pravimo tudi vgrajeni sistemi⁸. V vsakdanjem življenju se stalno srečujemo z dogodki, aktivnostmi in sistemi, ki dogodke in aktivnosti obvladujejo, od najbanalnejših, kot sta npr. vožnja avtomobila znotraj mesta ali opravljanje vsakodnevnih gospodinskih opravil, pa do najkompleksnejših in zahtevnih, kot sta npr. vodenje obratovanja nuklearne elektrarne ali upravljanje potniškega letala. Kot je mogoče že razvideti iz navedenih primerov, lahko hitro ocenimo, da vsi procesi niso enakovredni ne po zahtevnosti ne po odgovornosti in s tem resnosti posledic eventualnih anomalij v delovanju.

Slika 5: Sistem realnega časa



Vir: Gomaa, 2000, str. 8.

V kontekstu obvladovanja realnega časa sta tako bistvena pojma t.i. istočasnosti in pravočasnosti. Naslednja primera pojasnjujeta navedeno:

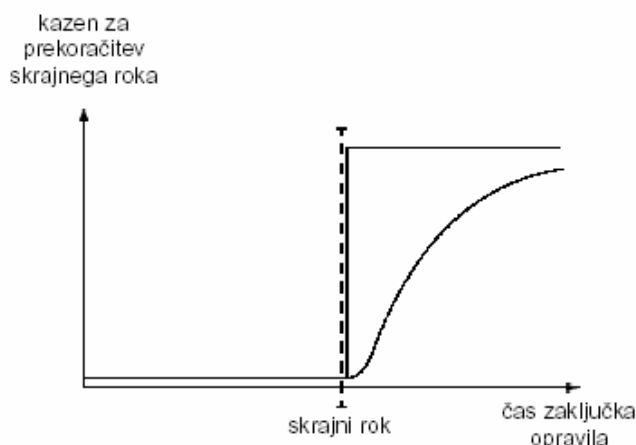
⁸ embedded systems

- Računalniški sistem za rezervacijo letalskih kart: četudi se opravilo ne zaključi v predvidenem času (npr. nekaj minut), bo lahko akcija še vedno uspešna (vozovnice rezervirane), le pri predolgem čakanju bodo stranke odšle drugam.
- Potnik gre na potovanje z letalom: če bo zaradi prometa na poti na letališče ali drugih vplivov potreboval več časa, kot je računal, bo letalo odletelo. Akcija ni uspešna.

Primeri kažeta, da je potrebno deliti sisteme v realnem času na tiste, pri katerih je mogoče tolerirati občasne zakasnitve, in na tiste, pri katerih zamuda termina pomeni neuspešno akcijo, ne glede na logično pravilnost rezultatov (čeprav je potnik prišel na letališče le minuto prepozno, je rezultat isti, kot da sploh ne bi bil prišel, ali bi bil šel na napačno letališče, kar pomeni, da je naredil logično napako).

Glede na navedeno ločimo dve okolji, in sicer okolje t.i. strogega realnega časa in t.i. mehkega realnega časa. Medtem ko pri drugem pri zakasnitvi roka kazenski stroški zakasnitve začnejo bolj ali manj hitro naraščati, pri prvem v trenutku skočijo na neko visoko vrednost.

Slika 6: Strogi in mehki realni čas



Vir: Colnarič, 2002, str. 2.

Da bo nek sistem dejansko deloval v strogem realnem času, mora izpolnjevati naslednje karakteristike:

- pravočasnost:** pri sistemih v realnem času srečamo dva časovna aspekta, in sicer relativnega, kjer se morajo akcije zgoditi v danem intervalu glede na nastop nekega dogodka, ter absolutnega, kjer je obnašanje sistema določeno s določenimi trenutki oziroma periodami, ko pričakujemo neko reakcijo.
- istočasnost:**⁹ pri sistemih v realnem času je mogoče istočasno pričakovati več, tudi hkratnih dogodkov, ki zahtevajo pravočasen odziv. Treba je zagotoviti, da bodo zahteve izpolnjene ne glede na to, koliko opravil se trenutno izvaja.

⁹ concurrence

- c) **napovedljivost:** da zadostimo zahtevam, moramo vnaprej poznati obnašanje sistema in s tem možnosti. Pomembno pri tem je obnašanje v najslabšem primeru, ki ga moramo predvideti (preobremenitev, izredne razmere..).
- d) **zanesljivost:** določa, koliko se lahko na sistem zanesemo. Pojem je sestavljen iz:
- pravilnosti: logična pravilnost ob nepravem času je prav tako napačna kot logična nepravilnost ob pravem času.
 - robustnosti: zahteva, da sistem ostane v specificiranem stanju tudi takrat, ko okolje ne ustreza več specifikacijam (npr. vhodne spremenljivke so izven predpisanih meja ipd.) ali pa ko izpade del sistema. Nujna je pravočasna detekcija izpadov, izkazana pa mora biti tudi trdoživost sistema. S pomočjo redundance, ki pomeni večkratno implementacijo posameznih delov sistema in preverjanje njihove konsistence, zagotovimo sprejemljivo obnašanje sistema tudi v prisotnosti programskih in aparaturnih napak.
 - razpoložljivost: procesi morajo zagotavljati neprekinjeno interakcijo z okoljem. Predčasni zaključki teh procesov zaradi napak niso dopustni.

V tem magistrskem delu se osredotočam na sisteme strogega realnega časa, kakršen je tudi sistem vodenja DEES znotraj kompleksa Distribucijskega centra vodenja.

3.2 Opis obstoječe konfiguracije DCV Elektro Primorska

Jedro obstoječega sistema daljinskega vodenja DCV predstavlja procesni računalniški sistem s podvojeno konfiguracijo (vroča rezerva) procesnih računalnikov PDP 11/44 in programskim paketom SCADA tip PROZA 11D/R, ki je proizvod podjetja Končar iz Zagreba. PDP 11/44 je DEC-ov srednje velik procesni računalniški sistem razvit ob koncu 70-tih let. Sistem vodenja predstavlja zaprti obratovalni sistem, ki ne omogoča standardnega prenosa podatkov v drugo računalniško okolje in tako ovira nadaljnji razvoj aplikacij vodenja in optimizacije procesa distribucije električne energije. Zaradi tega je bila z namenom čimvečje eksploatacije obstoječega sistema v času do implementacije novega izvedena "ad hoc" nadgraditev s programskim paketom PROZA OPEN, ki ima možnost povezovanja obstoječega SCADA sistema s poslovnim informacijskim sistemom v podjetju in omogoča posodabljanje določenih razvojnih in obratovalnih funkcij obstoječega sistema. Povezovanje obstoječega sistema vodenja v mrežo poslovnega informacijskega sistema pomeni nove obdelave funkcij vodenja in druge storitve možne na osnovi shranjenih podatkov in podatkov v realnem času. Dobljeni procesni podatki se iz PDP-jeve osnovne računalniške baze prenašajo v programsko PARADOX relacijsko bazo, ki deluje v WINDOWS okolju.

Nadgradnja s programskim paketom PROZA OPEN omogoča učinkovito in hitro obdelavo podatkov, ki smo jih v prejšnjem sistemu ročno pregledovali in obdelovali na izhodnih podatkovnih izpisnih listih. Danes poteka delo na osebnih računalnikih povezanih v informacijsko mrežo v standardnem Windows okolju in z grafičnimi vmesniki, kar omogoča obdelovalcu podatkov pravo orodje za njegovo analitično delo. Programski paket upošteva specifičnost pridobljenih podatkov iz sistema daljinskega vodenja za naloge preizkušanja, vzdrževanja in delovanja (enaki podatki ne pomenijo vedno istih dogodkov oziroma niso

vedno iz istih izvorov) in tako s pomočjo uporabnika pripomore h kvalitetnejšemu rezultatu izhodnih aplikacij.

3.2.1 SCADA procesni računalniški sistem

3.2.1.1 Strojna oprema v DCV

Jedro sistema daljinskega vodenja DCV predstavlja procesni računalniški sistem. Glavni deli računalniškega sistema so:

- 2x procesni računalnik DEC PDP 11/44,
- 2x komunikacijski podsistem,
- 2x konzolni tiskalnik,
- 3x tiskalnik,
- 4x upravljalno mesto INFOGRAF 3-2-1,
- 2x alfanumerični terminal,
- oprema za zvezo z daljinskimi postajami,
- sinhronizacijska ura.

Centralni računalniški del sestavljata dva procesna računalnika PDP 11/44, ki sta povezana v dualno konfiguracijo imenovano vroča rezerva. PDP 11/44 je DEC-ov srednje velik procesni računalniški sistem razvit ob koncu 70-ih let. Sistem operira s 16 bitno podatkovno besedo in 22 bitnim naslovnim vodilom. Računalnik je nameščen v DIGITAL-ov sistemski kabinet (ohišje) in je sestavljen iz sledečih modulov:

- centralne procesne enote s koprocesorjem ter 8kbyte cache spominom,
- 1MB delovnega spomina,
- SCSI disk kontrolerja ter disk enote 230Mb,
- kontrolerja za izmenljive diske ter 2x pogon za te disk enote 5Mb,
- 16 kanalnega RS232 kontrolerja,
- medračunalniške komunikacije 1Mb/s,
- ure realnega časa.

Komunikacijski DS-8 sistem omogoča izmenjavo podatkov z daljinskimi končnimi postajami¹⁰. Podsistem je preko pretvornikov tipa EDBC 11 povezan na UNIBUS zbiralnice procesnega računalnika PDP 11/44. Osnovna funkcija tega pretvornika je prilagoditev signala UNIBUS-a na zbiralnice DS8 sistema. Preslikanje zbiralnic omogoča, da računalnik vidi enote DS 8 sistema kot del svojega spomina. Komunikacijski podsistem vsebuje tudi nadzorno enoto EDBB155, ki služi za nadzor delovanja podsistema. V podsistemu so še I/O enote za povezavo lokalnih signalizacij v sistem daljinskega vodenja. Te signalizacije so delovanje komunikacijskega strežnika za oddaljene terminale, stanje sosednjega računalniškega sistema ter impulzi za sinhronizacijo točnega časa. Kot izhodni signal pa se uporablja izhod za aktiviranje zvočne signalizacije.

¹⁰ RTU - Remote Terminal Unit

V komunikacijskem podsistemu je vgrajeno 7 komunikacijskih enot, ki imajo vsaka svoj interni naslov. Vsaka omogoča pozivanje do 15 končnih postaj daljinskega vodenja. Komunikacija s končnimi postajami daljinskega vodenja je asinhrono serijska po principu master-slave oziroma poziv odgovor z grupiranjem podatkov po prioritetenih nivojih po licenci ASEA tip ADLP 80 (SINDAC).

Konzolni tiskalnik se uporablja pri zagonu in pri parametriranju računalniškega sistema ter kot terminal, na katerem se izpisujejo sistemska poročila, ki jih detektira diagnostika strojne in programske opreme. Konzolni tiskalnik je tip DEC LA 36.

Kot del človek-stroj¹¹ komunikacije se uporabljajo trije tiskalniki tipa DEC LA 100 za izpisovanje dogodkov, alarmov in okvar ter kopije zaslonских slik.

Lokalno upravljalno mesto tip INFOGRAF 3-2-1 je glavni operativni medij komunikacije operaterja s sistemom. Sestavljeno je iz kontrolne enote, 19" monitorja in funkcijske tipkovnice. Preko tega upravljalnega mesta ima operater nadzor preko celega sistema in s tipkovnico upravlja z njim. Tri taka mesta uporabljamo za vodenje elektroenergetskega sistema, četrto pa je namenjeno sistemskemu inženirju za vzdrževanje in projektiranje procesnega računalniškega sistema. Kontrolne enote INFOGRAF-a se nahajajo v posebni omari v računalniškem prostoru, ostala oprema pa je nameščena v komandni prostoru DCV.

Oprema za zvezo z daljinskimi postajami je nameščena v posebni omari in obsega napajalnike, modeme in kretnice ter služi za povezavo obeh komunikacijskih podsistemov s sistemom zvez. Za potrebe komunikacije z daljinskimi postajami uporabljamo modeme DS 8 sistema hitrosti 600 bit/s.

Sinhronizacijska radijska ura tip Funkuhr DCF 77 služi kot izvor točnega časa za sinhronizacijo časa v procesnem računalniškem sistemu. Sinhronizacija časa poteka na dveh nivojih in sicer prvič se pri zagonu procesnega računalnika avtomatsko odčita podatke iz radijske ure, kasneje pri delovanju pa se iz ure vodi sinhronizacijski impulz preko komunikacijskega podsistema v procesni računalnik. Radijska ura se sinhronizira glede na dolgovalovni oddajnik DCF77, ki je lociran v Meinfilingenu pri Frankfurtu in oddaja na frekvenci 77,5 kHz.

3.2.1.2 Programska oprema v DCV

Programsko opremo sistema vodenja tvorijo:

- operacijski sistem realnega časa RSX 11M,
- aplikativni program PROZA D/R,
- vzdrževalno programska oprema,
- pomožni in kontrolni programi.

Programska podpora implementirana v računalniški sistem sloni na programskem paketu PROZA 11 D/R za delo v realnem času (proizvod KONČAR). Proza 11 D/R je zasnovana na operacijskem sistemu RSX 11M. Operacijski sistem je večuporabniški in koncipiran tako, da hitro reagira v realnem času in hkrati omogoča razvoj programov za druge namene. Programski sistem PROZA 11 D/R omogoča reševanje problemov v realnem času. Real time del sistema podpira funkcije zbiranja, primarne obdelave, prikaza in upravljanja. Zahtevnejše

¹¹ HMI - Human Machine Interface

obdelave in funkcije planiranja se vršijo v razširjenem realnem času. Sistem podpira vključevanje različnih uporabniških programov. PROZA 11 D/R je izdelana iz neodvisnih programskih modulov. Kateri programski modul bo implementiran v sistemu se definira v generaciji sistema in je odvisen od zahtevanih funkcij. Modularnost sistema kot tudi mehanizmi vodenja skozi program omogočajo enostavno generiranje in inicializacijo sistema. Sistem je opremljen tudi z učinkovito diagnostiko.

Osnovne funkcije PROZA D/R sistema so:

- obdelava, prikaz in nadzor analognih, digitalnih in impulznih meritev,
- signalizacija vklopnih stanj naprav,
- signalizacija delovanja zaščit in ostalih enopoložajnih signalov v EES-u in SDV-ju,
- zvočno alarmiranje,
- kronološki zapis dogodkov,
- avtomatska izdelava poročil meritev,
- kratkoročno in dolgoročno arhiviranje informacij iz EES-a v smislu nadaljnje obdelave,
- daljinsko upravljanje krmiljenih naprav,
- distribucija točnega časa,
- nadzor sistema daljinskega vodenja (zbiranje podatkov in potrebne krmilne akcije),
- funkcije planiranja sistema (sprememba parametrov sistema).

Programski sistem definira niz jasno določenih pravil, po katerih se odvija komunikacija operaterja s sistemom. Ta pravila omogočajo obveščanje operaterja, prikaz podatkov in posege v proces. Operater komunicira s sistemom preko upravljalnih mest. Na zaslonih upravljalnega mesta ima vpogled v sistem in preko tipkovnice vpliva na njega. Dogajanja v procesu se dokumentirajo na tiskalnikih in tudi arhivirajo v datotekah. Sistem vsebuje tudi določene mehanizme zaščite pred napačnimi manipulacijami. Komunikacija se deli na več vrst dialogov z jasno deljenimi funkcijami. Vrste dialoga se delijo na dva podsistema, in sicer na elektroenergetski del EES in na sam sistem daljinskega vodenja SDV ter dva režima, in sicer vodenje sistema ter planiranje sistema.

3.3 Predstavitev in analiza operativno - delovnih procesov DCV

V okviru centra vodenja DEES se izvaja poslovni proces upravljanja distribucijskega omrežja (UDO), ki v osnovi vključuje vodenje DEES s funkcijami operativnega nadzora, odločanja in krmiljenja (SCADA¹² funkcije), analizo elektrodistribucijskega omrežja, vodenje postopkov vzdrževalnih in sanacijskih del ter obdelavo in pripravo podatkov za potrebe napovedovanja prevzema in odjema električne energije.

Iz teorije poznamo definicijo poslovnega procesa in delovnih procesov, ki so zbirka ene ali več aktivnosti, ki si sledijo v določenem vrstnem redu v izvajanju različnih nalog za izdelavo dobro definiranega izhoda (Damij, 2001, str.37). Delovni proces opredeljujejo vhodi, lastnik procesa, prevzemniki, omejitve, aktivnosti, dodana vrednost, strošek, čas, ključni dejavniki uspeha ter izhodi (Kovačič, 1998, str. 94). Poslovni proces kot nadrejeni nivo je torej primerna zbirka delovnih procesov (Damij, 2001, str.37). Ker je sam poslovni proces

¹² Supervisory, Control and Data Acquisition

upravljanja distribucijskega omrežja definiran kot zaključena celota, ki je predmet te naloge, nas na tem mestu ne zanima njegova vpetost z ostalimi poslovnimi procesi podjetja.

Analiza poslovnega procesa UDO in podrejenih delovnih procesov je bila v okviru elektrogospodarstva Slovenije izdelana v namensko izdelani študiji (Projekt integriranega modela podatkov, 1989, str. 182-200). Zaradi drugačne organiziranosti elektroenergetskega sektorja je bil takratni naziv predmetnega poslovnega procesa "Vodenje DEES". Izdelani analizi poslovnega procesa UDO oz. njegovemu operativnemu delu v sklopu DCV bom na tem mestu dodal še tiste delovne procese, ki so posledica sprememb poslovnega okolja in tudi možnosti, ki jih nudi sodobna informacijska tehnologija.

Najprej bom poslovni proces UDO opisal oziroma predstavil splošno po področjih, v nadaljevanju pa bom v smislu analize posamezna področja razdelal tako, da jih bo mogoče pri načrtovanju integralnega informacijskega sistema vključiti v fazo modeliranja procesov.

3.3.1 Splošna predstavitev področij UDO

Poslovni proces UDO opišemo po naslednjih področjih:

- **DISPEČIRANJE**, ki vključuje:
 - upravljanje pretokov električne energije po distribucijskem omrežju, upošteva tudi izmenjave energije z ostalimi povezanimi omrežji z vidika obratovanja,
 - skrb za varno in zanesljivo obratovanje distribucijskega omrežja,
 - zagotavljanje optimalne restavracije sistema po motnjah,
 - skrb za usklajeno delovanje distribucijskega omrežja,
 - izvajanje navodil o sistemskem obratovanju distribucijskega in prenosnega omrežja,
 - izvajanje obratovalnih in drugih navodil ter predpisov s področja obratovanja.
- **OPERATIVNO ENERGETIKO**, ki vključuje:
 - oblikovanje obratovalne statistike,
 - načrtovanje pretokov električne energije,
 - usklajevanje vzdrževalnih in drugih del z upravljalcem prenosnega omrežja (UPO) in distributerji,
 - izmenjevanje podatkov za varno obratovanje povezanih omrežij,
 - spremljanje števila in časa trajanja prekinitev za vse uporabnike,
 - obveščanje uporabnikov o prekinitvah dobave električne energije,
 - reševanje pritožb odjemalcev zaradi prekinitve dobave energije,
 - izdelovanje vozniških redov odjema oziroma porabe.
- **MERJENJE ELEKTRIČNE ENERGIJE**, ki vključuje:
 - skrb za številne meritve v distribucijskem omrežju,
 - usklajevanje merilnih podatkov prevzemno predajnih mest upravljalca prenosnega omrežja (UPO),
 - zbiranje in posredovanje podatkov odgovornim bilančnim skupin, bilančnih podskupin, Agenciji za energijo in upravljalcu prenosnega omrežja,

- zbiranje in posredovanje merilnih podatkov za obračun električne energije dobaviteljem in uporabnikom.
- **ZAŠČITA DEES IN IZVAJANJE OBRATOVALNIH MERITEV**, ki vključuje:
 - skrb za sistemske zaščitne naprave in lokalno avtomatiko,
 - skrb za obratovalne meritve v distribucijskem omrežju,
 - kreiranje zahtev in izdelovanje enotnih tehničnih karakteristik za naprave obratovalnih meritev in zaščite.
- **PROCESNO VODENJE**, ki vključuje:
 - skrb za tehnološki razvoj naprav procesnega vodenja,
 - skrb za vzdrževanje naprav procesnega vodenja,
 - kreiranje zahtev in izdelovanje enotnih tehničnih karakteristik za naprave procesnega vodenja.
- **KAKOVOST ELEKTRIČNE ENERGIJE**, ki vključuje:
 - skrb za meritve kakovosti napetosti,
 - skrb za delovanje sistema kakovosti napetosti v distribucijskem omrežju,
 - izdajanje izjav o skladnosti napetosti s predpisi.
- **VODENJE IN IZVAJANJE VZDRŽEVALNIH POSEGOV**, ki vključuje:
 - pripravo kriterijev in načrtovanje vzdrževalnih posegov,
 - izdelovanje predlogov vzdrževalnih posegov,
 - izdajanje soglasij k vzdrževalnim posegom,
 - nadziranje izvajanja vzdrževalnih posegov,
 - ugotavljanje mesta okvar in skrb za njihovo odpravo,
 - sodelovanje z uporabniki omrežja pri odpravi okvar.
- **INFORMACIJSKA PODPORA PROCESOM**, ki vključuje:
 - skrb za načrtovanje, razvoj in vzdrževanje informacijskega sistema,
 - vzdrževanje računalniškega sistema,
 - izdelovanje tehničnih navodil in navodil za uporabo,
 - parametriranje uporabniških aplikacij za zajem, obdelavo in shranjevanje podatkov.

Predmet obravnave tega magistrskega dela spada v zadnje navedeno področje, v okviru katerega se modelira in načrtuje informacijski sistem za podporo večini od ostalih področij.

3.3.2 Analiza delovnih procesov UDO

Ker obravnavam operativni del poslovnega procesa UDO, ki se izvaja v Distribucijskem centru vodenja, bom namesto izraza "poslovni proces UDO" uporabljal poenostavljen izraz "poslovni proces DCV".

Bistveni cilji, ki jih v okviru poslovnega procesa DCV zasledujemo, so (Projekt integriranega modela podatkov, 1989, str. 182):

- zbirati, prikazovati, nadzorovati, obdelovati in shranjevati podatke vklopnih stanj omrežja, merilnih veličin in stanja vseh gradnikov procesnega informacijskega sistema,
- operativno voditi spremembe vklopnega stanja omrežja in postrojev, regulirati napetost DEES, krmiliti porabo in omejevanje porabe električne energije,
- operativno planirati porabo in obremenitev omrežja na podlagi zbranih historičnih podatkov, izdelovati vozne rede obremenitev,
- analizirati obremenitve sistema, napetostne razmere, vklopna stanja, nastale dogodke v omrežju in postrojih ter obratovalno pripravljenost naprav v postrojih,
- izvajati izračune pretokov, obremenitev, izgub, kratkih stikov in napetosti na modelu omrežja na osnovi zbranih in v bazi podatkov shranjenih historičnih podatkov.
- lokalizirati mesta okvar v omrežju, voditi odpravo okvar in vzdrževalne posege v omrežju, pripravljati navodila za delo in preklope v omrežju, izvajati prenapajanja v omrežju in obveščati odjemalce o stanju v omrežju.

Iz navedenih ciljev lahko izluščimo sledeče delovne procese:

1. Nadzor obratovalnega stanja DEES
2. Vodenje obratovalnega stanja DEES
3. Planiranje porabe in obremenitev DEES
4. Izvajanje analiz obratovalnih stanj in dogodkov ter optimalnega obratovanja DEES
5. Izvajanje obdelave okvar v DEES

V nadaljevanju bom za delovni proces *Nadzor obratovalnega stanja DEES* izvedel njegovo funkcionalno dekompozicijo v aktivnosti in posle. Funkcionalna dekompozicija za ostale delovne procese je prikazana v prilogi tega dela.

Nadzor obratovalnega stanja DEES

AKTIVNOSTI	POSLO	OBSTOJEČA INFORMACIJSKA PODPORA
Zbiranje podatkov	• Registracija procesnih merilnih veličin iz nadzorovanih objektov. • Zapisovanje sprememb vklopnih stanj nadzorovanih objektov. • Kronološko zapisovanje alarmov in signalizacij iz nadzorovanih objektov, naprav za vodenje objektov, naprav TK zvez in ostalih pomožnih naprav. • Registracija podatkov o prevzeti električni energiji. • Zapisovanje ročno vnešenih podatkov o vklopnih stanjih in merilnih veličinah za objekte, ki niso daljinsko nadzorovani.	DA

Obdelava podatkov	• Obdelava surovih procesnih podatkov vklopnih stanj, meritev in signalizacij. • Izračuni moči, faktorja moči, obremenitev. • Izdelava poročil in tedenskih, mesečnih in letnih merilnih protokolov za moči, številne meritve in celotno EE bilanco. • Obdelava podatkov sistema vodenja.	DA
Prikaz podatkov	• Stalen prikaz vklopnih stanj, signalizacij in merilnih veličin na zaslonih prikazih. • Prikaz merilnih protokolov in poročil na zahtevo in možnost izpisa na tiskalnik. • Prikaz poročil o dogodkih, alarmih na omrežju, napravah in sistemu vodenja. • Prikaz in izpis poročil o obratovalnih in vzdrževalnih postopkih na omrežju. • Prikaz in izpis trenutnih topoloških shem, grafikonov obremenitev. • Prikaz geo-topologije omrežja na zahtevo, prikaz atributnih podatkov geo-elementov.	delno DA (zahtevana dopolnitev)
Arhiviranje podatkov	• Arhiviranje podatkov o dogodkih, alarmih, signalizacijah. • Arhiviranje podatkov o merilnih veličinah (obratovne meritve, meritve obremenitev, številne meritve). • Arhiviranje trenutnih stanj v DEES (celotna informacijska slika sistema).	delno DA (zahtevana dopolnitev)
Primerjava podatkov	• Primerjava trenutnih podatkov z referenčnimi vrednostmi (dopustne vrednosti, nastavljene vrednosti, normalna vklopna stanja, nastavitve pragov delovanja zaščit) • Primerjava arhiviranih podatkov z referenčnimi vrednostmi.	NE (zahtevana)

3.4 Modeliranje procesov in procesnih podatkov DEES

Avtomatizacijo in računalniško vodenje procesov v elektroenergetskih sistemih se je intenzivno pričelo uvajati ob koncu sedemdesetih let kot posledico uvajanja mikroprocesorskih računalniških sistemov. Prve generacije tovrstne strojne in programske opreme so že davno tehnološko zastarele, predvsem pa niso kos vedno večjemu številu podatkov, ki bi jih naj obdelale in predstavile za potrebe novih zahtev, ki jih prinaša spremenjeno poslovno okolje. Za implementacijo omenjenih klasičnih programskih sistemov je uporabljen t.i. funkcijsko-orientirani¹³ pristop k modeliranju procesnih podatkov. Le-ta je napram novemu, t.i. aparaturno-orientiranem pristopu¹⁴, s katerim se bom v nadaljevanju obširneje ukvarjal, manj učinkovit (Šimunić, 2001, str. 29-31). Učinkovitost drugega v primerjavi s prvim je v dejstvu, da je drugi v svojem bistvu objektno-orientirani¹⁵ pristop k modeliranju.

¹³ function-oriented

¹⁴ device-oriented

¹⁵ object-oriented

3.4.1 Upravljanje kompleksnosti procesov in podatkov

Operativno delo operaterja v DCV, katerega sistem vodenja je zgrajen na temelju omenjenega funkcijskega pristopa, bazira na analizi v DCV prispelih podatkovnih tokov. Na podlagi teh analiz operater sprejema odločitve in ustrezno ukrepa. Iz daljinskih končnih postaj prispeli podatki so operaterju predstavljeni kot dolge šifrirane liste logično nepovezanih signalov. Podatki pripadajoči neki napravi oziroma odgovarjajočemu polju kot funkcionalni celoti, nad katerim bdi ta naprava, so ob prispetju v sistem vodenja DCV razmetani preko več strani natisnjenih listov. S stališča operaterja predstavlja mnogo na tak način predstavljenih podatkov nekoristne, fragmentirane dele informacij, ki pa morajo biti s strani operaterja obdelane, memorizirane in korelirane z drugimi informacijami. Na tej osnovi si potem operater ustvari bolj ali manj jasno sliko dogajanja in stanja v tehnološkem procesu nekega elektroenergetskega postroja (EEP). Operaterjevo delo tako z rastjo in širitvijo primarnega elektroenergetskega sistema postaja izjemno kompleksna, zahtevna, predvsem pa odgovorna naloga, kar lahko postopoma prerase tudi v psihološki pritisk in posledično v povratno zanko, ki lahko generira napake v procesu odločanja. Predvsem take situacije nastajajo ob pojavu motenj v omrežju, ko je potrebno na dogodke nemudoma odreagirati.

Modeliranje elektroenergetskega sistema in procesov vezanih nanj se za potrebe izgradnje, kasnejše nadgradnje ter vzdrževanja informacijskega sistema vodenja DEES začne z modeliranjem procesnih podatkov. Struktura podatkov in podatkovnih tokov med daljinskimi končnimi postajami vodenja¹⁶, ki zajemajo in posredujejo podatke (komande, signali, alarmi, meritve), in centrom vodenja DEES mora biti poznana. Definiranje strukture signalov (podatkov) znotraj podatkovnih tokov je zahtevna naloga tudi za primer sistemov z relativno majhnimi perifernimi enotami in majhnim številom teh perifernih enot. Težave izhajajo iz sledečih dejstev:

- različnost implementirane opreme v EEP,
- kompleksnost tehnološkega procesa, ki presega znanje in izkušnje samo enega posameznika in je torej nujnost angažiranja skupine strokovnjakov za posamezen del tehnološkega procesa,
- relativno veliko število podatkov, ki se izmenjujejo med nadziranim in nadzornim sistemom vodenja DEES,
- potreba po velikem strokovnem znanju za specifiko posameznega EEP.

V nadaljevanju bom nakazal značilnosti funkcijskega pristopa pri izgradnji sistema vodenja DEES tako na nivoju modeliranja podatkov kot tudi na nivoju implementacije programskih rešitev. Omenjeni pristop je z obeh vidikov odvisen od uporabljene tehnologije in zato tudi preživet. Je pa zaradi razumevanja razvoja ter prednosti objektnega pristopa potrebno poznati tudi značilnosti funkcijskega pristopa, saj objektni delno temelji na funkcijskem.

¹⁶ RTU - Remote Terminal Unit

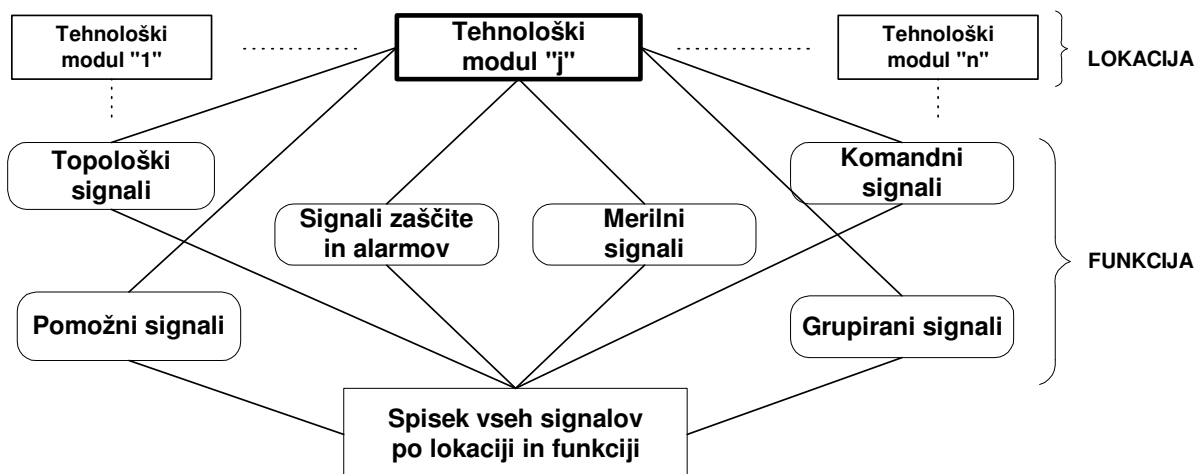
3.4.2 Funkcijski pristop

Tako funkcijsko-orientirana kot tudi objektno-orientirana paradigma sta v kontekstu modeliranja procesov in sistemov realnega sveta neločljivo povezani z obema istoimenskima paradigmama v svetu neposrednega računalniškega programiranja, ki je dejansko samo prenos modeliranih sistemov realnega sveta v modelirani informacijski sistem oziroma implementacija informacijskega sistema. V nadaljevanju bo prikazana funkcijska paradigma na področju modeliranja EEP.

3.4.2.1 Funkcijsko modeliranje procesnih podatkov

Funkcijski pristop je bil prva generacija v procesu modeliranja podatkov in ima svoje korenine v začetku osemdesetih let. Temelji na konceptu tehnoloških modulov znotraj tehnološkega procesa elektroenergetskega (EE) postroja in pripadajočih procesnih signalov razporejenih po njihovih funkcijah.

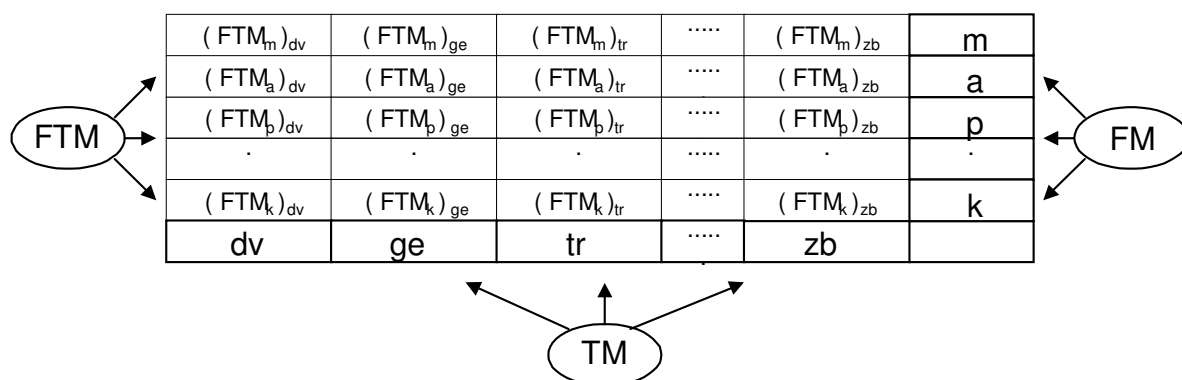
Slika 7: Funkcijsko orientirani model procesnih podatkov EEP



Vir: Šimunić, 2001, str. 31.

Procesni signali oziroma podatki so določeni glede na njihovo lokacijo, na katero je vezan pripadajoči tehnološki modul, in funkcijo, to je glede na njihov operativni pomen. Tehnološki moduli so vir in hkrati ponor signalov in v kontekstu modeliranja predstavljajo t.i. funkcijsko-tehnološke module (FTM), ki so osnova za definiranje strukture signalov (podatkov). Tabela 2 prikazuje FTM za primer različnih elementov DEES.

Tabela 2: Funkcijsko-tehnološki moduli

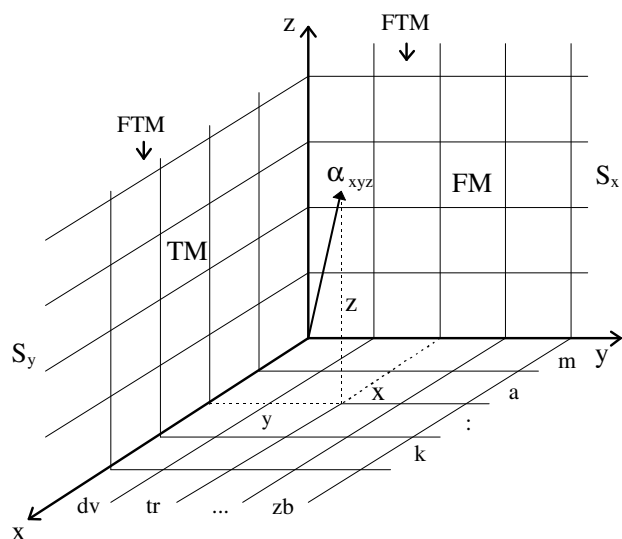


m - meritve dv - distribucijski vod FM - funkcijski modul
 a - alarmni signali tr - transformator TM - tehnološki modul
 p - položajni signali ge - generator FTM - funkcijsko-tehnološki modul
 k - komande zb - zbiranke

Vir: Šimunič, 2001, str. 31.

Ko poleg lokacije in funkcije vpeljemo še tretji aspekt signala (podatka) - vrednost signala, potem lahko odgovarjajočo podatkovno-signalno strukturo definiramo kot tridimenzionalni vektor (slika 8).

Slika 8: Funkcijska predstavitev signala kot vektorja



x - funkcija signala
 y - lokacija signala (tehnološki modul)
 z - vrednost signala
 α_{xyz} - signal definiran z lokacijo, kvaliteto in kvantiteto

Vir: Šimunič, 2001, str. 32

Nabor signalov S_{FT} lahko definiramo za vse tehnološke module in vse vrste signalov na nivoju EE postroja kot:

$$S_{FT} = \{\alpha_{xyz}\} \quad \text{kjer je} \quad \begin{aligned} x &= 1, 2, 3, \dots, m \\ y &= 1, 2, 3, \dots, n \\ z &= 1, 2, 3, \dots, r \end{aligned} \quad (1)$$

in α_{xyz} predstavlja signal definiran s funkcijo x , lokacijo y in vrednostjo z . Signalna oziroma podatkovna poddomena S_x predstavlja vse signale enega funkcijskega modula za vse tehnološke module EEP ob vrednosti $x = konst.$ Analogno predstavlja poddomena S_y vse signale vseh funkcijskih modulov za eden tehnološki modul EEP ob postavljeni vrednosti $y = konst.$ In končno poddomena S_{xy} (FTM na sliki 8) definira vse signale enega tehnološkega modula za en funkcijski modul ob postavitvi $x = konst.$ in $y = konst.$

Opisani model strukturiranja procesnih signalov na nivoju EE postrojev se je v preteklosti uporabljal za operativne analize sistemskih dogodkov in ocenjevanje različnih podatkovnih tokov. Kot takšnemu lahko pripišemo nekaj specifičnih omejitev in pomanjkljivosti kot npr. (Šimunić, 2001, str.33):

- Struktura procesnih signalov je organizirana na nivoju tehnoloških modulov EEP, zato je potrebno za vsak specifičen EEP ponoviti celotno proceduro modeliranja. Zgrajeni model ni neodvisen od specifik naprav RTP, prav tako modularno načrtovanje ni mogoče.
- Zaradi uporabe koncepta tehnoloških modulov naprava oziroma aparat kot temeljna komponenta ni samostojen element in ni obravnavana kot neodvisni vir podatkov.
- Stanja naprav ni mogoče definirati za potrebe modernega modeliranja in načrtovanja procesnih signalov, prav tako ni mogoče definirati povezav med napravami.
- Pri vnosu podatkov nastaja bistveno večja redundanca.

Kot primer v nadaljevanju prikazujem tabelo 3, kjer je opisan del podatkovne baze procesnih signalov, ki so modelirani na podlagi funkcijsko orientiranega pristopa.

Tabela 3: Funkcijsko-opisani procesni signali

20kV ZVEZNO POLJE													
FUN.	SIGNAL (NAZIV), KODA, VREDNOST					POROČILA							
						CV						RTP	
						D	A	O	KR	SH	S	KR	C
k o m.	Ločilnik	vklop	02200	3	xxx				+			+	+
	Ločilnik	izklop	02200	3	xxx				+			+	+
	Ločilnik	vklop	02200	3	xxx				+			+	+
	Ločilnik	izklop	02200	3	xxx				+			+	+
	Odklopnik	vklop	02200	3	xxx				+			+	+
	Odklopnik	izklop	02200	3	xxx				+			+	+
p o l	Ločilnik	vklopljen	02200	4	xxx	+			+	+	+	+	
	Ločilnik	izklopljen	02200	4	xxx	+			+	+	+	+	
	Ločilnik	medpoložaj	02200	4	xxx	+	+	+	+	+	+	+	

o ž a j. s i g.	Ločilnik	okvara	02200	4	xxx	+	+	+	+	+	+	+	+
	Ločilnik	vklopljen	02200	4	xxx	+			+	+	+	+	
	Ločilnik	izklopljen	02200	4	xxx	+			+	+	+	+	
	Ločilnik	medpoložaj	02200	4	xxx	+	+	+	+	+	+	+	
	Ločilnik	okvara	02200	4	xxx	+	+	+	+	+	+	+	
	Odklopnik	vklopljen	02200	4	xxx	+			+	+	+	+	
	Odklopnik	izklopljen	02200	4	xxx	+			+	+	+	+	
	Odklopnik	medpoložaj	02200	4	xxx	+	+	+	+	+	+	+	
	Odklopnik	okvara	02200	4	xxx	+	+	+	+	+	+	+	
za šči ta	Nadtokovna I>	delovala	02200	1	xxx				+			+	+
	Zemljskostična	delovala	02200	1	xxx				+			+	+
	Zaščita zbiralk 1	delovala	02200	1	xxx				+			+	+
a la rm i	Odklopnik	Air <16B	02200	2	xxx				+			+	
	Odklopnik	Air <14B	02200	2	xxx				+			+	
	Odklopnik	Air <12B	02200	2	xxx				+			+	
	Napajanje 220 V	okvara	02200	2	xxx				+			+	

Vir: Šimunič, 2001, str. 33.

Posamezni zapisi so razširjeni s kodirnimi parametri signalov in označenimi ciljimi lokacijami, kjer se določeni signali prikazujejo (tretji stolpec POROČILA). Prvi stolpec označuje funkcijo posameznega signala, kjer so signali grupirani glede na njihovo funkcijo. Drugi stolpec predstavlja naziv signala, kodo ter vrednost signala. V tretjem stolpcu so označene lokacije, kamor se signali razpošiljajo, bodisi v center vodenja bodisi na nivo postajnega računalnika v EEP. Poročila je možno razbrati iz različnih vrst list in shem npr. liste dogodkov (D), liste alarmov (A), liste okvar (O), liste kronoloških dogodkov (KR), enopolne sheme (SH), sinoptične sheme (S) ali lokalne sheme na visokonapetostni celici (C). Predhodno opisano tro-dimenzionalno kodiranje signalov temelji na sledečem principu:

- prva skupina digitov (dimenzija y opisanega modela - tehnološki moduli) se deli na tri podskupine:
 - prva dva digita označujeta vrsto polja npr. 02 - zvezno polje
 - naslednji digit označuje napetostni nivo polja npr. 2 - 20 kV
 - zadnja dva digita označujeta naziv polja npr. 00 - vezno polje 1
- druga skupina z enim digitom določa kvaliteto signala (dimenzija x opisanega modela), t.j. delitev signalov po funkciji kot npr. 3 - komande, 1 zaščite itd.
- tretja skupina treh digitov določa vrednost posameznega signala (dimenzija z opisanega modela).

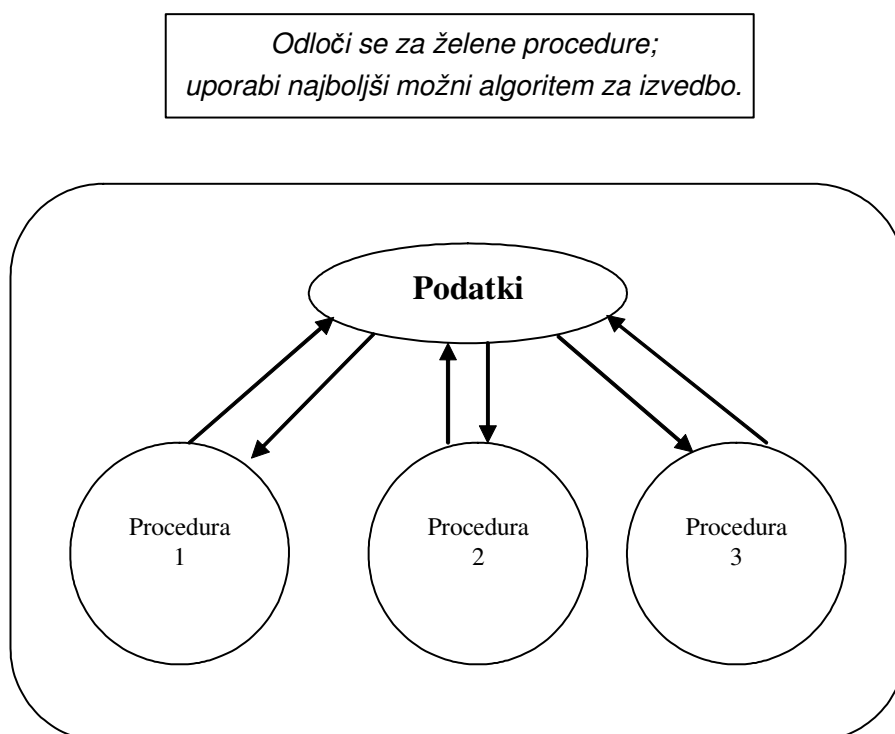
3.4.2.2 Proceduralno programiranje

Programska oprema je v zadnjih desetletjih postala nepogrešljiv del številnih tehničnih sistemov, med drugim tudi elektroenergetskih sistemov. Vsa ta oprema je bila razvita v okviru funkcijsko-orientirane paradigme oziroma v svetu računalništva bolj poznane proceduralnega programiranja. V veliki meri je bil izbor metodologije in implementacije odvisen od izbranega programskega jezika. Dobrobiti uporabe proceduralnih tehnik programiranja na področju izgradnje modelov elektroenergetskih sistemov sta bili sledeči (Soman, 2002, str. 8):

- analitični sistemski problemi so izraženi preko množice enačb (linearnih, nelinearnih, diferencialnih ipd.), s katerimi se proceduralni jeziki kot npr. Fortran ali C na tem nivoju abstrakcije zlahka spopadejo,
- dodobra razdelane tehnike numeričnih izračunov so v bistvu algoritmi, ki jih je mogoče enostavno implementirati v okviru proceduralnega programiranja.

Pri proceduralnem programiranju se torej osredotočimo na analitične in numerične metode ter algoritme za izvajanje želenih izračunov. Lahko govorimo, da proceduralno programiranje temelji na principu prikazanem na sliki 9:

Slika 9: Proceduralno programiranje



Vir: Soman, 2002, str. 8.

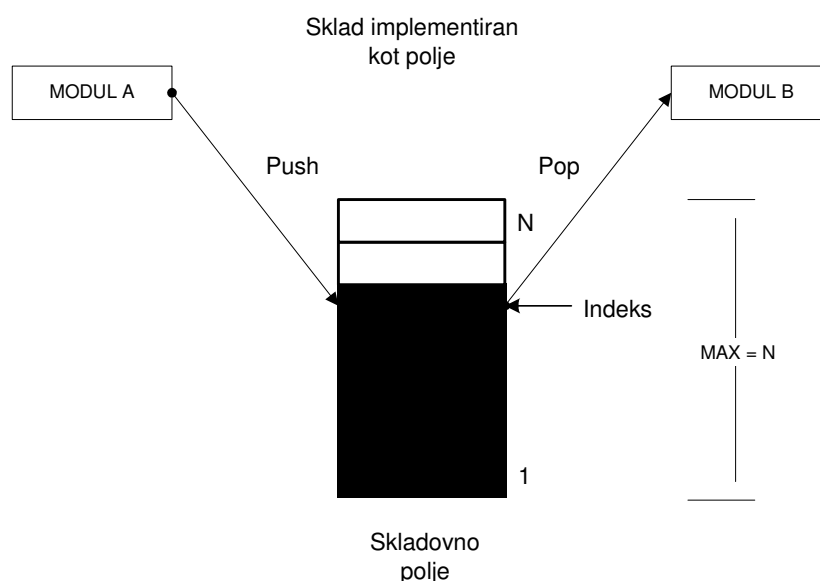
Slaba stran proceduralnega pristopa se izkazuje v dejstvu, da proceduralno programiranje ne poudarja abstrakcije¹⁴ in skrivanja (enkapsulacije)¹⁵ podatkov. Podatki so takorekoč »univerzalni«. Uporabljajo, delijo in spreminjajo jih procedure, ki se nanje navezujejo. Podatkovne strukture kot objekti in modeli realnega sveta so odprte. Povezav med podatki in njihovimi metodami ni, praktično vsakdo lahko preko vsake funkcije do njih dostopa, jih spreminja in dopolnjuje. Navedeno pomeni, da vsaka sprememba v podatkih in podatkovnih strukturah nujno pomeni spremembe in dopolnitve v procedurah in da je posledica vsake, še tako majhne, programske spremembe ali dopolnitve veliko število procedur ali funkcij, kar je za primer kompleksnih sistemov kot je npr. elektroenergetski zelo ranljivo in danes praktično neuporabno. V nadaljevanju tega dela bomo videli, da objektno-orientirani pristop znatno olajšuje in premaguje opisane pomanjkljivosti.

¹⁴ data abstraction

Za nazornejšo predstavbo navedenih pomanjkljivosti v nadaljevanju prikazujem uporabo oziroma neuporabo koncepta abstrakcije in skrivanja podatkov na primeru načrtovanja internih podatkovnih struktur (Gomma, 2000, str. 31-33).

Do pomnilniškega sklada¹⁶ dostopajo različni programski moduli, ki so v kontekstu funkcijskega programiranja procedure ali funkcije, v kontekstu objektnega, ki mu je osnova enkapsulacija, pa objekti. Pri funkcijskem pristopu je sklad globalna podatkovna struktura, do katere vsak modul dostopa direktno. Vsak modul mora torej za manipuliranje tabele, polja ali seznama kot skladovne podatkovne strukture poznati njegovo sliko oziroma predstavitev (slika 10).

Slika 10: Primer globalnega dostopa do skladovnega polja



Vir: Gomma, 2000, str. 32.

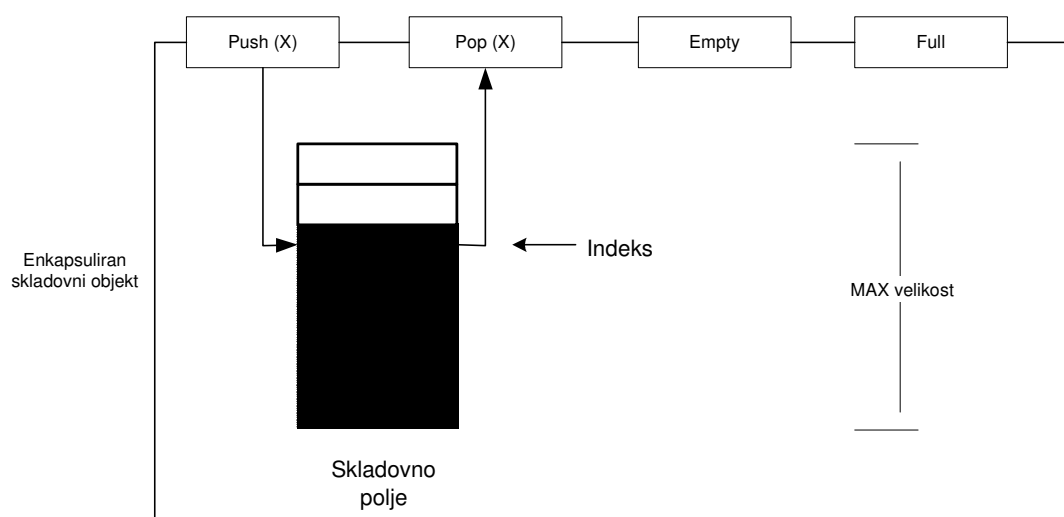
Rešitev z uporabo enkapsulacije vključuje skrivanje predstavitve sklada objektom, ki potrebujejo dostop do njega. V tem primeru je sklad predstavljen kot enkapsuliran objekt, načrtovan na sledeč način (slika 11):

- Za objekt sklad je definiran sklop operacij za manipuliranje podatkovne strukture kot npr. *push*, *pop*, *full*, *empty*.
- Definirana je podatkovna struktura sklada npr. enodimenzionalno polje. Dodatno sta definirana parameter za referenciranje naslova začetka skladovnega polja in parameter za definiranje velikosti skladovnega polja.
- Drugi objekti nimajo pravice direktnega dostopa do podatkovne strukture sklada. Dostop je omogočen le posredno preko klica operacije objekta (sklada).

¹⁵ data hiding (encapsulation)

¹⁶ stack

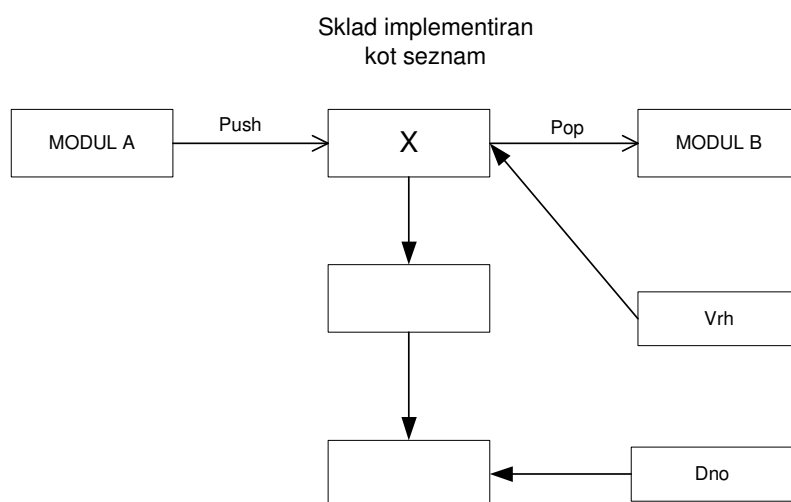
Slika 11: Primer skladovnega objekta – polja



Vir: Gomma, 2000, str. 33.

Predpostavimo, da spremenimo podatkovno strukturo sklada iz polja v povezano seznamsko podatkovno strukturo¹⁷ in raziščimo posledice za primer funkcijskega in enkapsulacijskega pristopa. V obeh primerih se mora spremeniti podatkovna struktura sklada. Pri funkcijskem pristopu je sklad implementiran kot globalna podatkovna struktura, zato se mora vsak modul, ki dostopa do sklada, spremeniti, saj modul dostopa do sklada oziroma do njegove podatkovne strukture direktno. Namesto manipuliranja z indeksi polja mora spremenjen modul manipulirati s kazalci povezanega seznama (slika 12).

Slika 12: Primer globalnega dostopa do skladovnega seznama

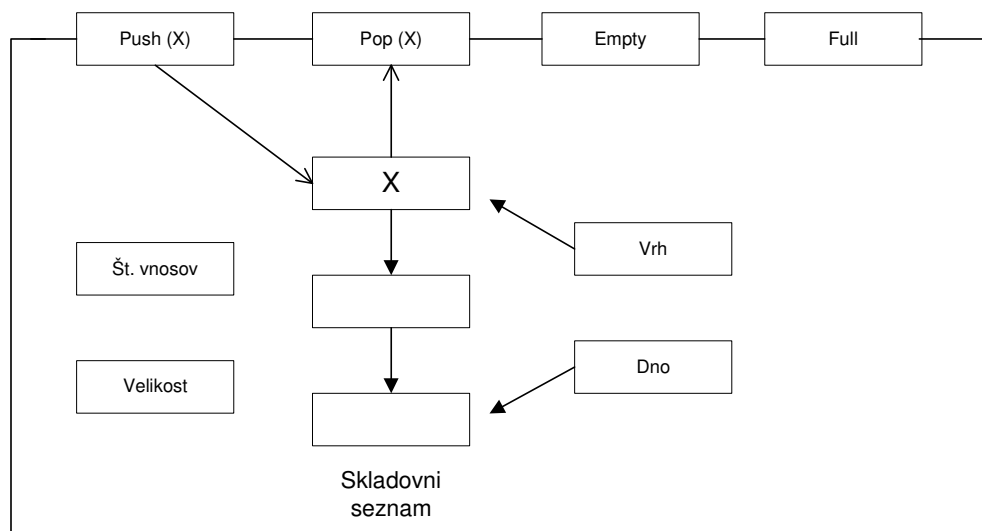


Vir: Gomma, 2000, str. 34.

¹⁷ linked list

V primeru objektnega pristopa se morajo zaradi spremembe podatkovne strukture sklada spremeniti notranje operacije objekta (sklada), saj sedaj dostopajo do povezanega seznama in ne skladovnega polja (slika 13). Kakorkoli, zunanji objektni (skladovni) vmesnik, ki je viden drugim objektom, ohrani vse svoje originalne lastnosti, torej se ne spremeni. Tudi zunanji objekti, ki uporabljajo sklad in do njega dostopajo le preko njegovih nespremenjenih vmesnikov, zato niso podvrženi kakršnimkoli spremembam. Le-ti nadaljujejo s klicem objektovih (skladovnih) procedur celo ne da bi vedeli za spremembe v klicanem objektu.

Slika 13: Primer skladovnega objekta – povezanega seznama



Vir: Gomma, 2000, str. 34.

3.4.2.3 Pomanjkljivosti funkcijskega pristopa

V zaključku tega podpoglavja želim poudariti, da je rezultat analize problemske domene nekega procesa generirani model, ki je abstraktna predstava nečesa z nekega specifičnega gledišča in zaobjema pomembne vidike modelirane stvari, poenostavlja pa ostalo, ki je pomembno z drugih gledišč. (Rumbaugh, 1999, str. 13-14). Model predstavlja sliko realnega sveta in je navadno preprostejši od realnosti, saj naj bi pripomogel k razumevanju kompleksnosti problemske domene. V tem smislu na tem mestu izpostavljam pomanjkljivosti tradicionalnega oziroma klasičnega pristopa, saj le-ta za analizo nekega procesa uporablja funkcije kot osnovne gradnike (bloke) za modeliranje problemske domene (Dilek, 2001. str. 16-17). Pri funkcijskem (proceduralnem) pristopu je način modeliranja nekega procesa ali sistema bistveno drugačen od načina uporabnikovega pogleda na problemsko domeno. Majhna sprememba v uporabnikovem pogledu oziroma modelu, t.j. v uporabnikovih zahtevah, lahko pomeni drastične spremembe v sistemskem modelu, t.j. programski kodi, kot je bilo navedeno v predhodnem tekstu.

Zaradi omenjenega je dandanes v uporabi objektno modeliranje in programiranje, saj omogoča lažje in natančnejše zajemanje zahtev in omogoča lažje sledenje spremembam v uporabnikovih zahtevah.

4 TEMELJNI KONCEPTI OBJEKTNEGA RAZVOJA INFORMACIJSKIH SISTEMOV

4.1 Splošno o informatizaciji poslovnih procesov

Večina podjetij delujočih na področjih gospodarstva, storitev, visoke tehnologije in tudi državne uprave je danes odvisnih od modernih informacijskih in nadzornih računalniških sistemov. Izkušnje pri snovanju in gradnji informacijskih sistemov so v obdobju zadnjih tridesetih let zadovoljive, čeprav danes v praksi srečamo tudi primere, ko sistemi ne izpolnjujejo v fazi priprave projektov zastavljenih ciljev. Razlogi za to so različni, med drugimi npr. previsoke ambicije, nejasni cilji, pomanjkanje operativnega obvladovanja procesov ter njihova neustrezna funkcijska specifikacija in končno tudi pomanjkanje same informacijske strategije¹⁸. Poleg tega se k snovanju IT projektov ni pristopalo integralno in je bilo v podjetjih tako apliciranih množica med sabo fizično in funkcionalno-logično ločenih sistemov ter baz podatkov. Med kardinalne napake lahko gotovo prištevamo tudi dejstvo, da se je zasnova in gradnja IT sistemov izvajala ne da bi pri tem upoštevali prilagajanje in prenovo samih bazičnih poslovnih procesov v podjetjih. Če je v upravljalških strukturah samo po sili razmer zavladalo prepričanje, da brez informatizacije poslovanja ni mogoče dosegati uspešnosti in učinkovitosti pri konkurenčnem boju na trgu, pa tega ne moremo reči za prilagajanje in prenovo poslovnih procesov. Ta posel je največkrat obveljal za »nebodigatreba« projekt, saj je posegal neposredno v delovne procese in s tem v taka in drugačna človeška razmerja. Danes izobraženi management sprejema dejstvo, da gresta prenova poslovnega procesa ter zasnova, razvoj in postavitve informacijskega sistema z roko v roki in tvorita osnovo informatizacije poslovanja.

4.1.1 Prenova poslovnih procesov

Podjetja na področju distribucije električne energije se danes soočajo z dvema velikima izzivoma. Prvi je v tej nalogi že omenjena deregulacija, drugi pa bliskovit razvoj informacijske tehnologije (Ottosson, 1998, str. 163). Medtem ko je deregulacija prinesla potrebo po strukturnih in operativnih spremembah v podjetjih vključujoč nove izzive in priložnosti, je razvoj IT področij podjetjem ponudil orodje za soočenje s temi izzivi ter obenem možnost za izkoriščenje novih priložnosti. Iz obeh dejstev nujno sledi zahteva po prenovljenem načinu poslovanja in globalni informatizaciji procesov.

Splošno gledano lahko prenovo poslovnih procesov opredelimo kot temeljito preverjanje procesov in njihovo radikalno spremembo za doseganje dramatičnih izboljšav kazalcev uspešnosti, kot so stroški, kakovost storitev in hitrost (Hammer, Champy, 1995, str. 50). Predstavlja večdisciplinarni in večfunkcijski pristop k prenovi, da lahko zadosti potrebam celotne organizacije. Vsebuje uskladitev kupčevih potreb s poslovnimi procesi za izločitev izmeta in stroškov, za skrajšanje proizvodnega cikla ter povečanje produktivnosti, ki ustvarja priložnost za doseganje poslovnih ciljev, tako rasti dobička kot tudi povečanja tržnega deleža (Watson, 1994, str. xi).

¹⁸ IT-strategy

Prenove procesov se lahko lotimo na bolj ali manj radikalen način. Poznamo (Kovačič, 1998, str. 99):

- celovito ali strateško prenovo poslovanja, ki je usmerjena v vsa ključna strateška vprašanja poslovanja podjetja in zajema v celoti vse poslovne procese podjetja in njihovo informatizacijo ter
- preureditev ali prenovo in informatizacijo posameznih poslovnih procesov ali njihovih delov. Pri tem gre največkrat poudarek možnostim, ki jih nudi sodobna informacijska tehnologija. Zato slednji obliki pravimo tudi informacijske prenova.

Vse bolj je danes razširjeno spoznanje, da mora podjetje, ki naj bo sposobno tekmovanja v svojem poslovnem okolju, obravnavati svoje poslovne procese celovito. Enako pomembno je ob vzpostavitvi celovito prenovljenih poslovnih procesov imeti razvit sistem nadzora nad njihovim izvajanjem. V kontekstu celovite obravnave in prenove poslovnih procesov lahko poslovne procese v podjetju delimo na ključne, podpirne in upravljalne (Črv, 2000, str. 59):

- Ključni poslovni procesi ustvarjajo vrednost z vidika kupcev in so tesno povezani s strateškimi usmeritvami poslovnega sistema. Rezultati teh procesov morajo imeti za odjemalce jasno razpoznavno vrednost.
- Podporni procesi neposredno ne ustvarjajo vrednosti za kupca, vendar ustvarjajo pogoje za delovanje ključnih poslovnih procesov. Njihov namen je predvsem zadovoljevanje notranjih potreb odjemalcev v ključnih procesih, z vidika kupcev pa lahko dodajajo vrednost le s podpiranjem ključnih procesov.
- Tretja skupina so upravljalni procesi, ki je njihov namen predvsem načrtovanje in nadziranje izvajanja preostalih dveh vrst procesov. Njihova pomembna naloga je tudi razvijanje in uveljavljanje ustreznega sistema meril za vrednotenje učinkovitosti poslovnih procesov in njihovih izvajalcev, posredno pa prispevajo k ustvarjanju in vzdrževanju ustrezne organizacijske kulture.

Za povečanje uspešnosti in učinkovitosti poslovanja je potrebno podrobno analizirati in po potrebi prenoviti vse tri vrste procesov. Ključni procesi morajo biti urejeni tako, da so usklajeni s potrebami kupcev. Posledično je zaradi spremembe ključnih procesov treba spremeniti tudi podpirne procese. Upravljalni procesi morajo vzpostaviti nov sistem meril za ocenjevanje novo oblikovanih procesov.

V literaturi (Hammer, Champy, 1995, str. 59-72) zasledimo, da je analiza obstoječih poslovnih procesov prvi korak v procesu prenove poslovanja podjetja. Prenovljeni poslovni procesi se potem bistveno razlikujejo od obstoječih, in sicer:

- **več nalog je združenih v eno.** Več specialističnih nalog združimo v eno kompleksnejšo. S tem izvajalec aktivnosti opravi več nalog, seveda pa mora biti zato ustrezno vsestransko usposobljen. Pri preveliki razdrobljenosti je oteženo koordiniranje, zaradi česar obstaja večja nevarnost napak, pa tudi časovni odzivi in mrtvi časi so večji;

- **izvajalci aktivnosti sprejemajo odločitve.** Hierarhična struktura se splošči. Nadzorni kader je zreduciran, stopnja zaupanja v izvajalce aktivnosti je večja, enako tudi odgovornost, saj izvajalec ni odgovoren le za izvedbo, ampak tudi za rezultat;
- **koraki v procesu se izvajajo v naravnem zaporedju.** Koraki si logično sledijo. Če je možno in smiselno, se lahko izvajajo vzporedno. Izvajanje procesov se s tem pospeši;
- **procesi imajo več različic.** Osnovni procesi so prilagojeni specifičnim zahtevam trga. Tako lahko govorimo o več različicah oziroma variantah enega procesa. Namesto enega zapletenega scenarija izvajanja procesa, ki vključuje vse možne zaplete, oblikujemo glavni proces, ki je kratek in učinkovit, ter alternativne procese v primeru zapletov. V toku dejanskega izvajanja procesa uvrstimo konkretno zahtevo kupca v ustrezen način obravnavanja. Za vsak primer uporabimo le tiste postopke, ki so nujno potrebni. Tako opredeljeni procesi so tudi bolj pregledni, saj vsaka varianta obravnava le primere, za katere je predvidena;
- **delo opravimo tam, kjer je to najbolj smiselno.** V tradicionalno organiziranih podjetjih delo opravljajo specialisti. Stroški poslovnega procesa se lahko bistveno znižajo in delo se opravi hitreje, če določene podporne funkcije (nabava, vzdrževanje ipd.) opravljajo sami izvajalci oziroma nosilci poslovnega procesa.
- **manj nadzora in preverjanja.** Novejši pristopi temeljijo na večjem zaupanju do sodelavcev. Ker je manj preverjanj, se procesi izvajajo hitreje. Pri tem pristopu dopuščamo zlorabe v razumnih in še sprejemljivih mejah oziroma izvajamo ekonomsko upravičen obseg kontrole;
- **napake so zmanjšane na najmanjšo možno mero.** S poenostavitvijo procesa in z zmanjšanjem števila dokumentov občutno zmanjšamo možnost, da bi prišlo do napak;
- **uporabljajo se hibridni centralizirano/decentralizirani procesi.** Proces je lahko centraliziran ali decentraliziran. Obe različici imata prednosti in slabosti. Izberemo organiziranost, ki je primernejša naravi procesa.

Po izvedeni prenovi podjetja se temeljito spremeni organiziranost podjetja. Spremembe lahko združimo v naslednjih točkah (Hammer, Champy, 1995, str. 73-89):

- klasične funkcionalne oddelke nadomestijo procesne skupine
- delo se iz preprostih opravil spremeni v večrazsežno delo
- izvajalci aktivnosti niso več nadzorovani, ampak dobijo pooblastila
- spremeni se način izobraževanja
- nagrajevanje je odvisno od rezultatov
- napreduje se glede na sposobnosti in ne glede na rezultate
- spremeni se organizacijska kultura podjetja
- managerji niso več nadzorniki, ampak postanejo mentorji
- hierarhična struktura se splošči
- najvišji management se približa izvajalcem aktivnosti in kupcem

Zaradi vseh navedenih sprememb je pri prenovi poslovnih procesov vedno treba računati na odpor znotraj podjetja npr. v organizacijskih enotah, ki se jih prenova dotika. Predvsem je močan odpor managerjev na srednjem in nižjih nivojih, ki v pričakovanih spremembah vidijo nevarnost za svoj položaj in se ne čutijo sposobne za nov način opravljanja dela.

Zato se je spreminjanja poslovnih procesov treba lotiti na način, ki bo pojasnil potrebnost predvidenih sprememb. Eden od možnih je uporaba naslednjega postopka za spreminjanje procesov, ki je v uporabi tudi v okviru programov postopnih izboljšav, kot je celovito obvladovanje kakovosti¹⁹.

V prvem koraku je treba povečati pripravljenost ljudi na spremembe. To izvedemo tako, da preučimo obstoječ način dela in razumemo kritične dejavnike uspeha, zahtevane zmožnosti in primanjkljaj potrebnih sposobnosti. Treba je jasno opredeliti, kakšen naj bo bodoči proces. Opredelitev prihodnosti mora biti jasna in razumljena s strani vseh udeležencev, ki se jih sprememba dotika. V procesu vzpostavljanja skupnega razumevanja problema izločimo večino strahu pred spremembami.

Drugi korak zahteva dokumentiranje novih poslovnih procesov. Ob dokumentiranju se navadno pokažejo prej skrite nedoslednosti in prekrivanje vlog izvajalcev. Nejasnosti je treba podrobno opredeliti in proces dokumentirati na urejen način npr. v okviru sistema ISO 9001.

V tretjem koraku poskušamo izboljšati produktivnost z zmanjševanjem stroškov, ki jih lahko enostavno identificiramo. Izločimo tiste aktivnosti, ki ne dodajajo vrednosti in tako poskušamo čimbolj poenostaviti proces.

V četrtem koraku izvedemo nadaljnjo optimizacijo z zmanjševanjem variabilnosti. Z uporabo poskusnega načrtovanja ter statističnega spremljanja rezultatov procesa poskušamo oblikovati proces, ki konsistentno izdeluje izdelke, ki med seboj ne odstopajo. Šele ko smo končali četrti korak izločevanja nepotrebnih aktivnosti, lahko proces učinkovito avtomatiziramo.

Pri prenovi poslovnih procesov je tveganje tem večje, čim bolj je proces pomemben in čim večji del organizacije je treba prenoviti. Ob tem lahko verjetnost neuspeha še povečamo, če se prenove ne lotimo dovolj celovito in ne zagotovimo potrebnih predpostavk. Najpomembnejša predpostavka je odločenost managementa v podjetju, da se prenova opravi, ter dosledna vztrajnost na izbrani poti. Potrebno je zagotoviti vse potrebne pogoje za realizacijo, še posebej se to nanaša na ustrezno informacijsko tehnologijo in zagotavljanje potrebnih kadrovskih virov. Pri prenovi je treba upoštevati obstoječo kulturo podjetja z razumevanjem, kako jo je potrebno spremeniti, da lahko prenova uspe. V kolikor ne spremenimo kulture in načina definiranja dobrega rezultata, je malo verjetno, da bodo prenovljeni procesi zaživel, kot je predvideno v načrtu prenove. Pri samem postopku preoblikovanja procesov se ne smemo zadovoljiti z malimi popravki, če je možno s korenito spremembo doseči bistveno boljše rezultate.

4.1.2 Pristopi k zasnovi in razvoju informacijskih sistemov

V procesu preoblikovanja podjetja ima informacijska tehnologija ključno vlogo. Brez ustrezne informacijske tehnologije si prenove sploh ne moremo predstavljati. To pa še ne pomeni, da že sama informacijska tehnologija pomeni uspeh v procesu prenove podjetja. Informatizacija ob starem načinu razmišljanja in dela lahko stanje celo poslabša. Da bi z uvajanjem informacijske tehnologije dosegli zelen uspeh, moramo, kot že omenjeno, istočasno izvesti prenovo procesov. Pri tem je informacijska tehnologija stimulator teh sprememb (Watson,

¹⁹ angleško: total quality management, TQM

1994, str. 129). V končni fazi integracija prenove poslovnih procesov in aplikacije informacijskih tehnologij prinaša večjo fleksibilnost poslovnih procesov.

V sklopu strateškega načrtovanja razvoja podjetja je danes strateško načrtovanje razvoja informatike njegov bistveni element (Kovačič, 1998, str. 42-62). Informatika je eno od ciljnih področij v strateškem poslovnem načrtu za zagotavljanje konkurenčnih prednosti podjetja. Odvisnost podjetij od informatike je velika. Govorimo lahko o obdobju strateških informacijskih sistemov. Strateški informacijski sistemi so sistemi za zagotavljanje konkurenčnih prednosti in za izoblikovanje tekmovalne strategije podjetja.

Za strateško načrtovanje informatike je poznanih več pristopov:

- **tradicionalni pristop.** Temu pristopu pravimo tudi parcialni pristop, saj problematike informatike ne zajema celovito. Zadovoljimo se s posameznimi programskimi rešitvami ali s posameznimi uporabniškimi aplikacijami;
- **celoviti pristop z vrha navzdol.** Ta pristop je naprednejši, saj poskuša informatiko obravnavati celovito. Pomanjkljivost tega pristopa je kompleksnost in nepreglednost kot posledica zahtevnosti obravnavanega problema;
- **postopen razvoj informatike s sprotim preverjanjem rezultatov.** Je pristop, ki poskuša odpraviti pomanjkljivosti predhodnih metod. Na eni stani prevelike razdrobljenosti in na drugi strani prevelike kompleksnosti. Je večfazni pristop z vrha navzdol. Pristop je zasnovan na sprotne preverjanju vseh faz in na istočasnem dograjevanju dobljenih rezultatov (Kovačič, Vintar, 1994, str. 165). Odlika tega pristopa je v tesni povezanosti razvijalca in uporabnika informacijskega sistema. Posamezno fazo opredelimo le do stopnje natančnosti, ki je potrebna za dobro razumevanje. S tem se izognemo preveliki kompleksnosti.

Računalniške zmogljivosti so v preteklosti predstavljale veliko omejitev pri uvajanju informatizacije v poslovanje podjetij. Težišče raziskovanja je zato bilo osredotočeno na iskanje učinkovitih algoritmov, ki naj bi optimalno izkoriščali razpoložljive zmogljivosti v okviru izdelanih programskih aplikativnih rešitev. Take programske rešitve so bile po vsebini omejene, izdelovali pa so jih programerji specialisti. Njihova znanja so bila predvsem tehnične narave, manj pa so poznali vsakodnevne poslovne probleme, s katerimi so se praviloma srečevali potencialni uporabniki izdelanih rešitev. Poleg tega so jim pogosto manjkale tudi komunikacijske sposobnosti, saj tovrstna usposabljanja še niso bila dognana kot potrebna. Uporabniki so tako imeli težave pri prenašanju svojih potreb programerjem, ti pa z razumevanjem praktičnih problemov uporabnikov (Avison, Fitzgerald, 1996, str.18). Temu primerno so se pojavljali problemi:

- rešitve velikokrat niso zadovoljevale uporabniških zahtev,
- četudi je bila rešitev ustrezna, ker je programer zelo dobro poznal problem in strukturo rešitve, je bilo izvajanje sprememb v kolikor toliko obsežni rešitvi dolgotrajno in zamudno zaradi posledic, ki so se lahko pojavile v drugem delu sistema,
- rešitve so bile pogosto slabo dokumentirane.

Iz navedenih razlogov so podjetja kmalu postala odvisna od posameznikov, ki so podrobno poznali strukturo rešitev, saj so bile le-te za druge programerje nepregledne in neuporabne zaradi pomanjkanja dokumentacije ter uporabe različnih tehnik programiranja.

Kot odgovor na navedene probleme se je pojavila nadaljnja delitev dela pri izgradnji informacijskih sistemov:

- analiza poslovnih zahtev, ki naj jo opravi sistemski analitik,
- načrtovanje sistema, ki naj izpolni definirane poslovne zahteve,
- programiranje in implementacija.

Tako se poleg same implementacije pojavita še vsaj dve pomembni fazi v razvoju informacijskih sistemov, in sicer analiza ter načrtovanje. Naslednja reakcija na opisane probleme je pojav formaliziranih metodologij ter orodij za uporabo metodologij v procesu razvoja računalniških aplikacij (Avison, Fitzgerald, 1996, str. 20).

4.1.3 Metodologije razvoja informacijskih sistemov

V literaturi zasledimo več definicij za metodologije razvoja in izgradnje informacijskih sistemov. Ena izmed njih le-te opredeli za integrirano zbirko organizacijskih procedur in tehnik ter avtomatskih podpornih orodij, ki z uporabo sistematičnega pristopa omogočajo in olajšajo razvoj, implementacijo ter vzdrževanje nekega procesa informacijskega sistema (Minio, 2001, str. 1). V praksi si pod tem pojmom največkrat predstavljamo organizacijska in tehnična znanja, ki so uporabljena pri snovanju in izdelavi računalniško podprtih rešitev (Kovačič, Vintar, 1994, str. 29).

Namera sistematične uporabe metodologije je zagotoviti hiter odziv na porajajoča se vprašanja in dileme kot npr. kaj in kdaj je potrebno narediti, kdo naj pri tem sodeluje, katere so kontrolne točke procesa in kako zagotoviti ustrezno kvaliteto izvedbe. Ciljna in sistematična uporaba metodologij na področju poslovanja podjetij za distribucijo električne torej prinaša dobrobiti, med drugimi za management definira strukture in tehnike planiranja, potrebe za proračun, potrebe za obratovanje, nadzor in vzdrževanje primarnega sistema. Občutno se zmanjšajo razvojni in implementacijski časovni okvirji, omogočeno je spremljanje napredovanja projekta v smislu obvladovanja kakovosti (TQM). Sama metodologija se ponuja tudi kot orodje za komunikacijo med uporabniki in projektno skupino za razvoj in izgradnjo informacijskega sistema. Uporabnikom je obenem tudi dana možnost, da se aplikacije prilagajajo njihovim potrebam, in tudi spodbuda, da se lahko aktivno vključijo v projekt v celotni njegovi življenjski dobi.

Natančneje lahko cilje, ki jih z uporabo metodologije lahko dosežemo, strnemo v sledeče (Avison, Fitzgerald, 1996, str. 11):

- Natančno beleženje informacijskih zahtev. Uporaba metodologije pomaga pri opredeljevanju in analizi uporabniških zahtev²⁰.
- Podana je sistematična metoda razvoja tako, da lahko napredek učinkovito nadzorujemo. Nadzor je posebnega pomena pri projektih velikega obsega, zato so nujne kontrolne

²⁰ angleško: Requirements analysis

točke in dobro definirane faze, ki omogočajo uporabo tehnik načrtovanja in vodenja projektov.

- Uporaba metodologije omogoča razvoj informacijskega sistema v primernem časovnem okviru in ob sprejemljivih stroških.
- Izdelan informacijski sistem je dobro dokumentiran in enostaven za vzdrževanje. Potreba po prihodnjih spremembah sistema je neizbežna zaradi sprememb v organizaciji in poslovnem okolju. Spremembe so izvedljive z minimalnim vplivom na preostale dele sistema.
- Zagotovljena je indikacija potrebnih sprememb v zgodnji fazi razvojnega procesa. Strošek spremembe se namreč povečuje, ko izgradnja sistema napreduje od začetne faze analize proti fazi implementacije sistema.
- Izdelan informacijski sistem je prilagojen ljudem, ki so z njim v interakciji. Tako je zagotovljena uspešnost uporabe takega informacijskega sistema.

V nadaljevanju sta navedena primera klasičnih metodologij, ki naj bi vsaj deloma zasledovale zgoraj navedene cilje.

4.1.3.1 EPT metodologija²¹ (Minio, 2001, str. 1-5)

Edenor je argentinsko podjetje za distribucijo električne energije. Podjetje je bilo soočeno z novimi izzivi deregulacije sistema. Dotedanja informatizacija je potekala ločeno za različna področja poslovanja podjetja in na različne načine. Nekatere rešitve so bile kupljene kot končni proizvodi specializiranih podjetij, druge so bile razvite s strani zunanjih razvijalcev po kvazistandardnih vzorcih, tretje pa so bile namensko »po meri« izdelane za potrebe nekaterih poslovnih funkcij podjetja. Tipični primeri področij takšnega načina aplikacij informacijskih sistemov so:

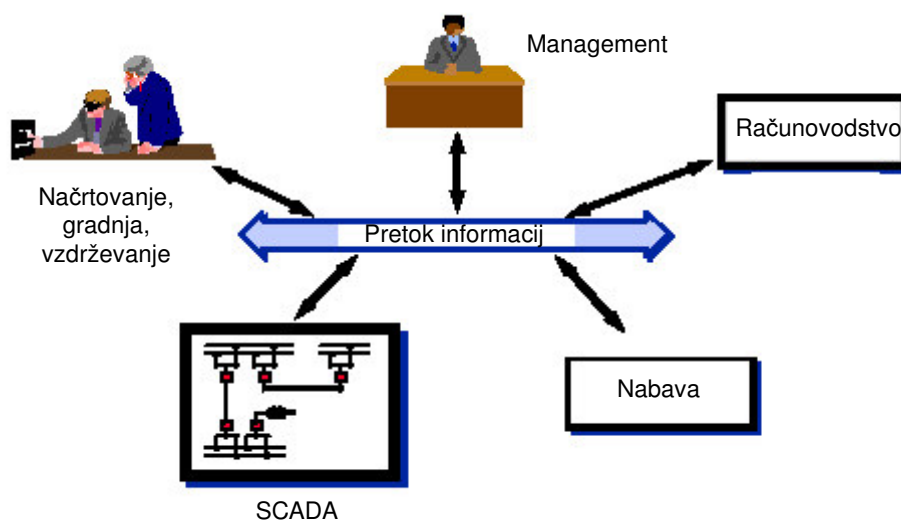
- tehnični informacijski sistem vključujoč bazo tehničnih podatkov o odjemalcih in napravah, prostorske podatke o lokaciji naprav in tehnične kalkulacije za potrebe razvoja DEES ter CAD²² aplikacije,
- administrativna področja vodenja podjetja (finance, računovodstvo, kadri itd.),
- računalniško podprta orodja za vodenje DEES npr. SCADA/DMS orodja.

Poleg tega, da je prevladalo spoznanje, da se zahteve okolja hitro spreminjajo in je zato potrebno obvladati pristope, ki bodo tem spremembam lahko sledili ter da je nujno vzpostaviti informacijsko povezavo med do tedaj ločenimi informacijskimi otoki, je k oblikovanju nove strategije načrtovanja informatike in metodologije prispevalo tudi dejstvo, da tudi parcialni projekti niso nikoli zaživel tako, kot je bilo prvotno zamišljeno. Zato je metodologija strukturirana tako, da omogoča osredotočenje na stalno spremljanje kvalitete procesa izgradnje informacijskega sistema, ki bo pokrival vse poslovne zahteve. Primer končne faze take integracije in izgradnje informacijskega sistema prikazuje slika 14.

²¹ Edenor Systems Development Methodology

²² Computer Aided Design orodja

Slika 14: IT sistem sodobnega elektro podjetja



Vir: Ottosson, 1998, str. 166.

Metodologija je organizirana v hierarhično strukturo, ki vsebuje faze²³, fronte²⁴, etape²⁵ in naloge²⁶.

- **Faze** so veliki sklopi aktivnosti znotraj cikla razvoja, ki se jih sekvenčno razvija, in vsaka od njih zahteva formalno potrditev, preden se lahko prične z naslednjo.
- **Fronte** odgovarjajo specialističnim področjem znotraj projektnega tima, ki je multidisciplinaren. To pomeni, da so vanj vključeni strokovnjaki z različnih področij.
- **Etape** so posamezne homogene naloge združene v projektni fazi. Tako združevanje omogoča lažje spremljanje poteka projekta. Za posamezne etape znotraj neke faze ni nujno, da se izvajajo v določenem zaporedju.
- **Naloge** predstavljajo skupino relativno homogenih aktivnosti, ki se prilagajajo projektnim etapam.

Faze in fronte se v večji meri dotikajo poslovno-organizacijskih aspektov izvajanja projekta, medtem ko etape in naloge neposredno posegajo v samo operativno izvajanje projekta.

Metodologija razločuje med sledečimi fazami:

- Faza planiranja informacijskega sistema
- Faza osredotočenja in definiranja projekta
- Faza projektnega planiranja
- Faza systemske analize in načrtovanja
- Faza implementacije
- Faza post-implementacijske podpore in vzdrževanja

²³ phases

²⁴ fronts

²⁵ stages

²⁶ tasks

Znotraj vsake faze nastopajo štiri fronte oziroma specializirana področja, ki so vključena v izvajanje projekta:

- **Administrativna fronta.** Ustreza administraciji projekta in je skupna aktivnost vsem fazam. Izvajanje in kontrola projektnega dela, zaključevanje in prevzemanje projekta ter kapitalizacija pridobljenih izkušenj so del teh aktivnosti.
- **Aplikacijska fronta.** Je bistveni del aktivnosti znotraj posamezne faze, med drugimi sistemsko in projektno planiranje ter razvojne in vzdrževalne aktivnosti.
- **Fronta planiranja.** Ustreza specifičnim aktivnostim, ki so povezane z izdelavo potrebnih planov za implementacijo določenih rešitev znotraj ostalih front. Tu se pripravljajo tudi nekatere kompleksne naloge kot npr. usposabljanje zaposlenih, izvajanje testiranj, pretvorba podatkov itd.
- **Operativna fronta.** Ustreza aktivnostim povezanim z operativno organizacijo, kjer se sistemsko planiranje, razvoj in vzdrževanje izvajajo.
- **Tehnološka fronta.** Ustreza aktivnostim povezanim z definiranjem, strategijo in aplikacijo informacijske tehnologije za podporo poslovanju podjetja.

Projektni tim za izvajanje projekta mora biti sestavljen z upoštevanjem specialističnih področij znotraj vsake fronte, da bi tako lahko identificiral in izvajal etape in naloge v posameznih fazah projekta.

Znotraj vsake faze izvajanja projekta se v okviru posameznih front izvajajo etapne naloge. Na tem mestu bi torej sledil konkretnější in podrobnejši prikaz globlje strukture metodologije, a bi to presegalo okvir tega dela.

EPT metodologija daje velik poudarek na dokumentiranje izvajanja in rezultatov vsake faze projekta, in sicer:

- Faza planiranja informacijskega sistema: Izdelano in izdano je končno poročilo o sistemskem planu izvajanja projekta ter strateških usmeritvah.
- Faza osredotočenja in definiranja projekta: Izdelana je detajlna projektna dokumentacija.
- Faza projektnega planiranja: Izdelan dokument, ki podrobno obdeluje konceptualno načrtovanje informacijskega sistema, ter operativno-terminalni plan del.
- Faza sistemske analize in načrtovanja: Izdelan je dokument s funkcionalnim opisom, tehničnim opisom in podrobnejšimi sistemsko-tehničnimi specifikacijami.
- Faza implementacije: Dokumentirane so programske specifikacije, specifikacije podatkovne baze, testni model, implementacijske procedure, pretvorba podatkov, priročniki za uporabo in vzdrževanje itd.
- Faza post-implementacijske podpore in vzdrževanja: Dokumentirajo se zahteve, predlogi izboljšav in pritožbe, periodični sistemski pregledi in revizije itd.

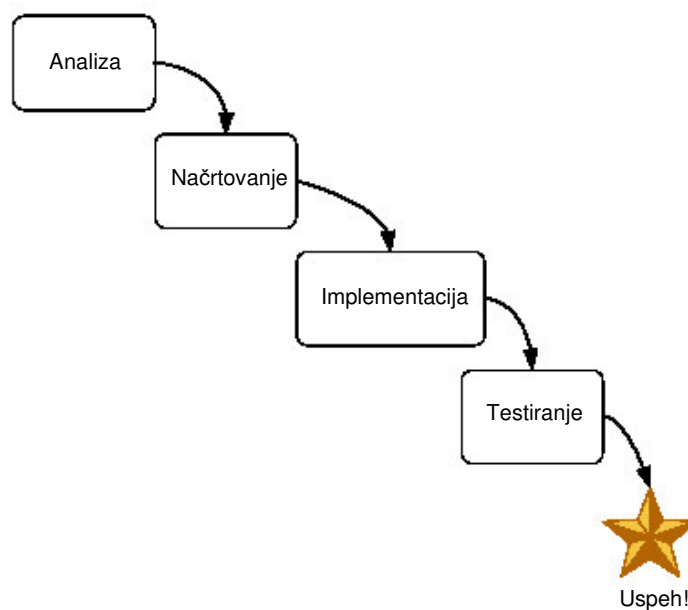
Vsa dokumentacija se mora ažurno aktualizirati glede na spremembe, kar urejajo posebej za to izdani dokumenti.

4.1.3.2 Metodologija življenjskega cikla sistema²⁷

Metodologija je nastala ob koncu šestdesetih let v Veliki Britaniji in predstavlja osnovno izhodišče mnogih kasneje razvitih metodologij. Že sama sintagma »življenjski cikel« napeljuje na kasneje nadgrajeno metodologijo iterativnega procesa razvoja informacijskih sistemov. Zaradi sprememb v organizaciji in okolju se dopolnjujejo uporabniške zahteve, kar sproži nov cikel razvoja oziroma prenove informacijskega sistema.

Metodologija življenjskega cikla sistema temelji na striktno zaporednem izvajanju razvojnih faz izgradnje informacijskega sistema. Produkti razvoja znotraj posamezne faze morajo zadostiti določeni stopnji »zrelosti«, preden se lahko prične izvajati naslednja faza. Metodologija vključuje naslednje faze razvoja (slika 15):

Slika 15: Model življenjskega cikla sistema



Vir: Douglass, 1999, str. 7.

Praviloma se pred začetkom izvajanja projekta preliminarно pripravi študija izvedljivosti, ki analizira obstoječi informacijski sistem, nove informacijske zahteve, probleme obstoječega sistema in nove tehnološke možnosti. Pripravijo se alternativne rešitve, ki se ocenijo po pravno formalni, organizacijski, tehnični in ekonomski plati. Opredelijo se meje sistema, potrebni viri in opredelijo stroški in koristi posameznih alternativ. Izbere se najboljša rešitev, ki se posreduje vodstvu v obliki formalnega poročila. Odobritev poročila s strani vodstva in odločitve o nadaljevanju projekta predstavljata tako ničelno kontrolno točko metodologije.

Ko je nadaljevanje projekta odobreno s strani vodstva, je potrebno v fazi raziskovanja sistema podrobno analizirati zahteve uporabnikov in pripraviti funkcionalno specifikacijo sistema. Potrebno je podrobno opredeliti tipe in količino podatkov, omejitve ter probleme obstoječega sistema. Informacije pridobimo s pomočjo intervjujev vodilnega in operativnega kadra, z uporabo vprašalnikov, analizo obstoječih dokumentov in opazovanjem načinov dela.

²⁷ The Waterfall Lifecycle Methodology

Orodja za prikazovanje rezultatov so diagrami poteka, hierarhični diagram strukture poročanja v organizaciji, specifikacije obstoječih dokumentov, zapiski intervjujev ipd. Z nadaljnjo analizo odkrivamo vzroke obstoječih problemov, razloge za uporabo določenih metod in obstoj alternativnih metod, predviden porast količine podatkov ter poskušamo razumeti razloge za trenutno stanje sistema in nakazati načine za izboljšanje.

Fazi analize sledi faza načrtovanja sistema, v kateri je treba narediti načrt bodočega sistema. Potrebno je opredeliti podatkovne strukture, načine zajemanja podatkov, procese za transformacijo vhodnih podatkov v izhodne, procedure za varovanje in restavriranje podatkov. Opravi se odločitev o izboru konkretnega okolja implementacije, v smislu izbora strojne in systemske programske opreme ter sistema za upravljanje z bazami podatkov, če ga uporabimo. Postopkovni del se dokumentira v obliki diagramov poteka, mrežni diagram se uporabi za prikaz povezav med procesi in podatki oziroma kateri programi uporabljajo določene podatke.

Sledi faza implementacije, ki vsebuje nakup in inštalacijo opreme, kodiranje in testiranje programov ter kreiranje dokumentacije. Nadalje je potrebno izvesti integralni test sistema tudi s strani uporabnikov ob uporabi večje količine (realnih) podatkov, da preizkusimo tako funkcionalno kot tudi performančno ustreznost sistema. Izvesti je treba še šolanje uporabnikov sistema in izvesti prehod iz starega na novi sistem.

Ko sistem že operativno deluje, se izvede pregled sistema, kjer se preveri skladnost delovanja sistema z opredeljenimi zahtevami in primerjava dejanskih stroškov s predvidenimi. Pripraviti je potrebno tudi formalno poročilo. V okviru vzdrževanja se odpravljajo v sistemu prisotne napake ter dograjujejo funkcionalnosti, ki so potrebne zaradi novih zahtev uporabnikov ali spremenjenega okolja. V kolikor so potrebne spremembe prevelike, da bi jih lahko izvedli brez večjih posegov v sistem, se sproži nov cikel razvoja sistema.

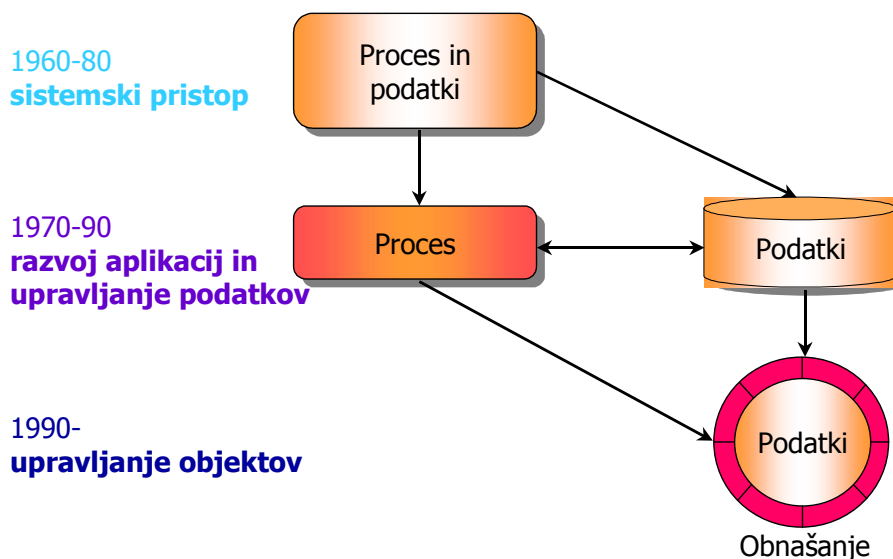
Prednost opisane metodologije se kaže v tem, da je zaradi linearnosti procesa možno enostavno predvidevati, načrtovati in razporejati naloge. Pomanjkljivosti pa se pokažejo predvsem v tem, da za razvojne produkte posameznih faz ni mogoče zagotoviti končne dovršenosti. To pomeni, da je pomanjkljivosti mogoče identificirati šele takrat, ko se produkt ene faze prenese v elaboracijo v naslednjo. Izkušnje kažejo, da analiza ne more biti zaključena dokler ni bilo opravljenega vsaj nekaj načrtovanja, prav tako faza načrtovanja ni zaključena dokler se ne prične izvajati kodiranje ter da kodiranje (implementacija) ne more biti zaključeno preden se prične testiranje. V klasičnem življenjskem ciklu je uspešnost izvedbe vsake faze odvisna od definitivnosti in natančnosti produktnega rezultata predhodne faze, kar se pa v praksi nikoli ne dogodi. V tem smislu je opisana metodologija nedovršena in neprimerna. Samo projektno planiranje in izvedba sta sicer olajšana, medtem pa so sami plani pomanjkljivi in v splošnem ni mogoče nikoli z gotovostjo reči, na kateri točki izvedbe se projekt trenutno nahaja.

4.2 Objektni pristopi k razvoju informacijskih sistemov

Objektni pristopi k razvoju informacijskih sistemov vključujejo predvsem tehnike in metodologije modeliranja procesov in podatkov realnega sveta. Objektne metodologije modeliranja se bistveno razlikujejo od drugih pristopov, saj združujejo tisto, kar je bilo do sedaj nenaravno ločeno: modeliranje podatkov in modeliranje procesov. Metodologije razvoja informacijskih sistemov pred pojavom objektno orientiranih metodologij namreč dosledno ločujejo podatkovni in procesni pogled. Tako se podatki in procesi modelirajo v različnih modelih, v fazi implementacije pa se podatki shranjujejo v podatkovni bazi, proces pa se kodira v programske procedure in rešitve, ki manipulirajo s temi podatki. Pri tem nastanejo težave pri spremembah podatkovne strukture, ki obvezno povzroči iskanje programskih modulov, ki uporabljajo spremenjeno strukturo (sliki 16 in 17).

Najpomembnejši cilj razvoja in implementacije tovrstnih sistemov je zmanjšanje stroškov in zmanjšanje časa izgradnje informacijskih sistemov. Pri gradnji in prenovi informacijskih sistemov potrebujemo orodja, s katerimi bomo v kratkem času sposobni razviti kakovostne informacijske sisteme. Objektni pristop poizkuša objekte realnega sveta preslikati v objekte objektnega modela. Model tako postane abstrahirana in poenostavljena realnost, ki je narejena za boljše razumevanje sistema ali procesa, ki ga modeliramo (Booch, Rumbaugh, Jacobson, 1999, str. 463). Stopnja abstrakcije realnega sistema (stopnja konkretizirane implementacije modela) je odvisna od namena neke faze modeliranja sistema. Tako se npr. izvaja modeliranje na višjem nivoju²⁸, v katerega so vključeni praviloma vsi udeleženci pri izgradnji informacijskega sistema in obsega zgodnje osredotočenje na sam modelirani proces, na zbiranje zahtev in predvidevanje opcij. V kasnejši implementacijski fazi se izvaja modeliranje na nižjem nivoju²⁹, v katerega so vključeni neposredni razvijalci programske opreme in obsega detajlno kodiranje funkcionalnosti in aplikacij (Rumbaugh, Jacobson, Booch, 1999, str. 15).

Slika 16: Razvoj upravljanja podatkov

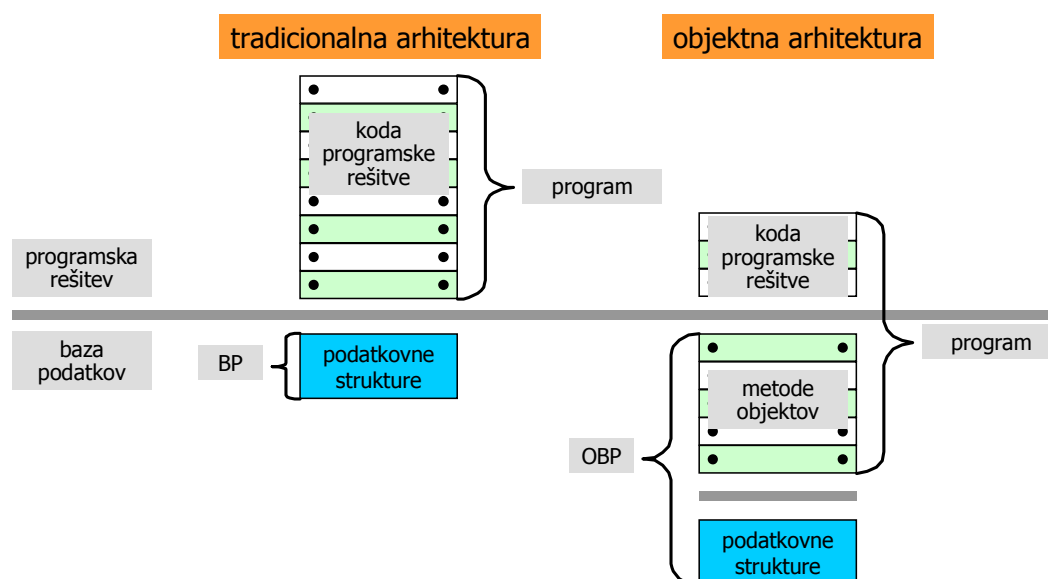


Vir: Baze podatkov – študijsko gradivo, 1999.

²⁸ High-level

²⁹ Low-level

Slika 17: Arhitektura: program – baza podatkov



Vir: Baze podatkov – študijsko gradivo, 1999.

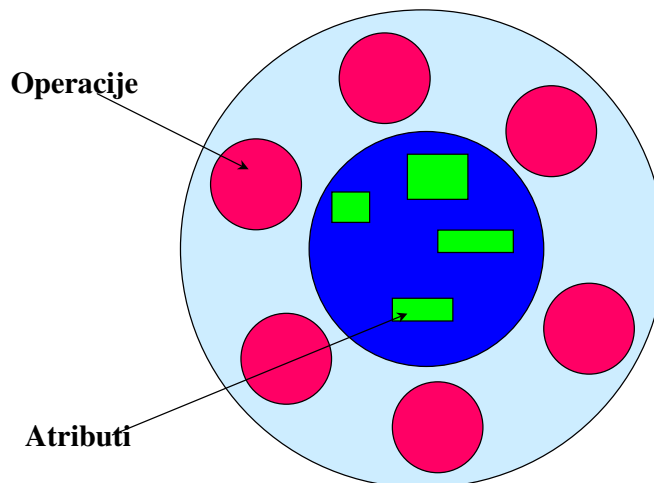
Objektno orientirana analiza in objektno orientirana konstrukcija omogočata boljše razumevanje problemov in boljše sodelovanje vseh zainteresiranih. Poveča se konsistentnost med analizo, načrtovanjem ter konstrukcijo in samim objektno orientiranim kodiranjem. Z analizo in načrtovanjem opišemo, kaj bomo delali, s konstrukcijo, kako bomo delali, s kodiranjem pa implementiramo programsko aplikacijo. Objektni pristop omogoča lažje spreminjanje sistema, kar je posledica večkratne uporabnosti atributov in operacij ter tudi večkratne uporabnosti rezultatov analize, konstrukcije in kodiranja. Glede na klasične funkcionalno-strukturne pristope si od objektno orientiranega pristopa obetamo povečanje produktivnosti, povečanje kakovosti izvedenih projektov, poenostavitev vzdrževanja in poenostavitev nadgradnje.

4.2.1 Osnovni pojmi objektno orientiranih pristopov

Objekt je katerakoli stvar (entiteta) realnega sveta, realna ali abstraktna, o kateri hranimo podatke in njeno obnašanje, to je operacije nad njenimi podatki (Grad, Jaklič, 1996, str. 117-122; Martin, 1993, str. 17-33). **Objektni tip** je **kategorija objekta**. Objekt pa je **primerek** objektnega tipa.

Objekt je v modelu predstavljen z vrednostmi svojih **atributov** (svojstev) in z opisom **obnašanja**. Atribut je imenovana lastnost razreda, ki opisuje razpon vrednosti, ki ga primerki lastnosti lahko vsebujejo. Predstavlja neko lastnost stvari, ki jo modeliramo, ki je skupna vsem objektom razreda (Booch, Jacobson, Rumbaugh, 1999, str. 50).

Slika 18: Predstavitev koncepta objekta



Vir: Baze podatkov – študijsko gradivo, 1999.

Opis obnašanja objekta je v modelu predstavljen z **operacijami**, ki se uporabljajo za branje ali manipulacijo podatkov o objektu. Operacija je implementacija storitve, ki jo lahko drugi objekti zahtevajo od razreda z namenom vplivanja na obnašanje (Booch, Jacobson, Rumbaugh, 1999, str. 51). Operacije omogočajo komunikacijo med objekti. Objekti ne morejo neposredno dosegati podatkov o drugih objektih. Če jih želijo dosegati, jim morajo poslati **zahtevo**. Zahteva povzroči, da se operacija nad enim ali več objekti izvede. Zahtevi pravimo tudi **sporočilo**. Operaciji, ki je implementirana v programu, pravimo **metoda**. Vsak objekt je enolično določen s svojim identifikatorjem, ki je ločen od njegovega stanja, kar pomeni, da imata lahko dva objekta isto stanje, a različni identiteti.

Ograjevanje³⁰ je lastnost objekta, da skrije podatke pred ostalimi objekti. Ostali objekti lahko dosegaajo njegove podatke le preko njegovih operacij. Ograjevanje ščiti objekte pred izgubo in neželenimi manipulacijami z njihovimi podatki. Zunanji uporabniki (drugi objekti) vedo, kaj operacija naredi z objektom, ne vedo pa kako. Koncept skrivanja podatkov³¹ vodi do nizke sklopljenosti med objekti, saj omeji število domen, o katerih mora uporabnik objekta razmišljati. Za uporabnika je pomembno le to, katere operacije so mu na voljo, katere informacije te operacije pričakujejo in katere informacije bodo vrnjene uporabniku. Med objektom - odjemalcem in klicanim objektom - strežnikom je na ta način vzpostavljena in osnovana pogodba. Ta pogodba jasno definira in opisuje dialog oziroma način komuniciranja med njima.

Razred je implementacija **objektnega tipa**. Z **generalizacijo** tipu določimo **nadtip** in s **specializacijo** **podtip**. Na ta način dobimo hierarhijo tipov. Razred vsebuje podatkovne strukture in metode, ki opredeljujejo operacije, ki jih lahko uporabimo nad podatki. Hierarhija razredov tako omogoča **dedovanje**³². V primeru, ko imamo v modelu dva ali več razredov s podobnimi atributi in operacijami, lahko s postopkom generalizacije izločimo skupne attribute

³⁰ Encapsulation

³¹ Data hiding

³² Inheritance

in operacije v poseben razred, ki ga z uporabo mehanizma dedovanja povežemo z razredi, ki ohranijo le tiste lastnosti, ki so jim lastne in jih tako izdelani nadrazred nima. Obratno lahko s postopkom specializacije iz razreda, ki ima ustrezne lastnosti, kreiramo podrazred, ki mu dodamo posebne operacije ali attribute, ki jih potrebujemo. **Podrazred** tako deduje podatkovne strukture in metode **nadrazreda**, vsebuje pa lahko tudi lastne podatkovne strukture in metode. Včasih prav z namenom kreiranja novih podrazredov kreiramo t.i. **abstraktni razred**, ki ima le funkcijo združevanja lastnosti, ki jih dedujejo njegovi potomci. Tak razred navadno nima primerkov, imajo jih le njegovi potomci. Znotraj specialnega razreda lahko poleg dodajanja atributov in operacij spremenimo tudi pomen in implementacijo določene operacije. Postopku pravimo predefiniranje. Postopek je zelo učinkovit, vendar nam poslabša preglednost celotnega modela, saj na prvi pogled ni vidna različna funkcionalnost operacije z enakim imenom v dveh z mehanizmom dedovanja povezanih razredih. Potomec lahko deduje lastnosti od enega ali več prednikov. V prvem primeru govorimo o enkratnem, v drugem pa o večkratnem dedovanju. Večkratno dedovanje prav tako zmanjšuje preglednost modela in jo je treba posebej previdno uporabljati. Ob nepravilni uporabi lahko pride do ponavljajočega se dedovanja. Dedovanje ob pazljivi uporabi omogoča učinkovito vzdrževanje, saj se spremembe operacij vršijo na enem mestu, na najvišjem mestu v hierarhiji razredov. Tako spremenjeno operacijo podedujejo vsi potomci tega razreda. Dedovanje omogoča obsežno ponovno uporabo, saj se z dedovanjem lastnosti izbranega razreda in dodajanjem potrebnih funkcionalnosti bistveno skrajšuje čas implementacije.

Asociacije predstavljajo povezave med primerki razredov. Na konceptualnem nivoju predstavljajo asociacije konceptualne povezave med razredi (Fowler, Scott, 2000, str. 52). Asociacija je strukturna povezava, ki pove, da so objekti ene stvari povezani z objekti druge (Booch, Rumbaugh, Jacobson, 1999, str. 65). Asociacija med seboj lahko povezuje dva ali redkeje tudi več razredov. V prvem primeru jo imenujemo binarna asociacija, v drugem primeru pa n-terna asociacija. Grafično jo prikazujemo kot polno črto, ki povezuje razreda med seboj. Asociaciji lahko priredimo ime in smer asociacije. Asociacija je lahko eno ali dvosmerna in predstavlja relacijo med razredi. Binarne asociacije prikažemo s črto, ki povezuje simbola dveh razredov. S pomočjo različnih načinov prikazovanja črt lahko predstavimo različne značilnosti asociacije. N-terne asociacije prikažemo s pomočjo romba, ki je povezan z razredi vključenimi v asociacijo. Asociacijo običajno poimenujemo, ob imenu pa lahko s pomočjo majhnega črnega trikotnika določimo tudi smer branja. Razen tega lahko asociacijam določimo še števnost, vloge, določila, navigacijo in omejitve. Osnovne vrste asociacij predstavljajo naslednje strukture:

- »je-povezan-z« (angleško: is-associated-with),
- »je« (angleško: is-a).
- je-del« (angleško: is-part-of),

Prva predstavlja običajno obliko povezave, druga pa je tesno povezana s pojmom generalizacije in specializacije v okviru koncepta dedovanja. Zadnja predstavlja relacijo, ki se imenuje **agregacija** ali združitev in je izrazita oblika asociacije, pri kateri je združiten objekt sestavljen iz komponent. Komponente so deli (»part - of«) agregata. Ta je semantično

razširjen objekt, ki je obravnavan kot enota pri mnogih operacijah, čeprav je fizično sestavljen iz več manjših objektov. Posebna oblika agregacije je kompozicija, ki je v bistvu močnejša oblika agregacije. Poenostavljeno velja, da se pri kompoziciji npr. operacija »izbriši« posreduje in prenese tudi do vseh sestavnih delov. Razlika med kompozicijo in agregacijo je torej v tem, da je kompozicija posebna vrsta agregacije s poudarjenim lastništvom.

Mnogoličnost ali polimorfizem je sposobnost, da iste metode uporabljamo nad različnimi razredi. Pri tem se različni razredi različno odzivajo. Prilagodljivost oziroma fleksibilnost sistema se s tem principom bistveno poveča. Kadar ima operacija z istim imenom v različnih razredih različen pomen in implementacijo, govorimo o mnogoličnosti. Na način izvedbe torej vpliva razred prejemnika sporočila. Sporočilo lahko sproži različen odziv glede na razred, v katerem je klicana operacija implementirana. To tudi pomeni, da je ista operacija implementirana na več različnih načinov oziroma zanjo obstaja več metod.

Ločimo tri tipe polimorfizma (Osnovni OO koncepti, 2005):

- Vključitveni polimorfizem³³, ki je posledica tega, da lahko eno samo ime predstavlja objekte različnih razredov, ki so povezani z nekim skupnim nadrazredom. Dedovanje je zato ključ do te oblike polimorfizma. Z vključitvenim polimorfizmom je tesno povezana tehnika dinamičnega povezovanja (dynamic binding). To je proces, ki omogoča, da se identifikacija natančno določene metode odloži vse do časa izvajanja.
- Operacijski polimorfizem³⁴ je tip polimorfizma, ko se z istim imenom sklicujemo na objekte, ki ne izkazujejo kakšne skupne povezave v smislu podrazredov oz. razredne hierarhije dedovanja. Oblika operacijskega polimorfizma je t.i. "function overloading" polimorfizem.
- Parametrski polimorfizem³⁵ pogosto imenujemo tudi splošnost (genericity). Glavna značilnost je, da pri deklaraciji splošnega tipa ali razreda kot parametre uporabimo tipe.

4.2.2 Poenoteni objektno orientirani modelirni jezik UML

UML³⁶ je splošno uporabljiv vizualni modelirni jezik za razvoj sistemov (Arlow, Neustadt, 2002, str. 5). Vsaka inženirska panoga potrebuje standarden jezik, ki inženirjem in tudi uporabnikom omogoča nedvoumno komunikacijo. Pri razvoju programske opreme smo priča pestri množici notacij, med katerimi preslikava ni vedno možna. Veliko število razvojnih tehnik in mnoge bolj ali manj podobne notacije so v razvoj programskih sistemov vnesle dvoumnost. Največja prednost jezika za objektno modeliranje UML je prav poenotenje komunikacije v svetu programskega inženirstva. Največja podjetja in njihova orodja danes podpirajo jezik UML, kar pomeni, da lahko uporabniki pričakujejo enostavno izmenjavo

³³ Inclusion polymorphism

³⁴ Operation polymorphism

³⁵ Parametric polymorphism

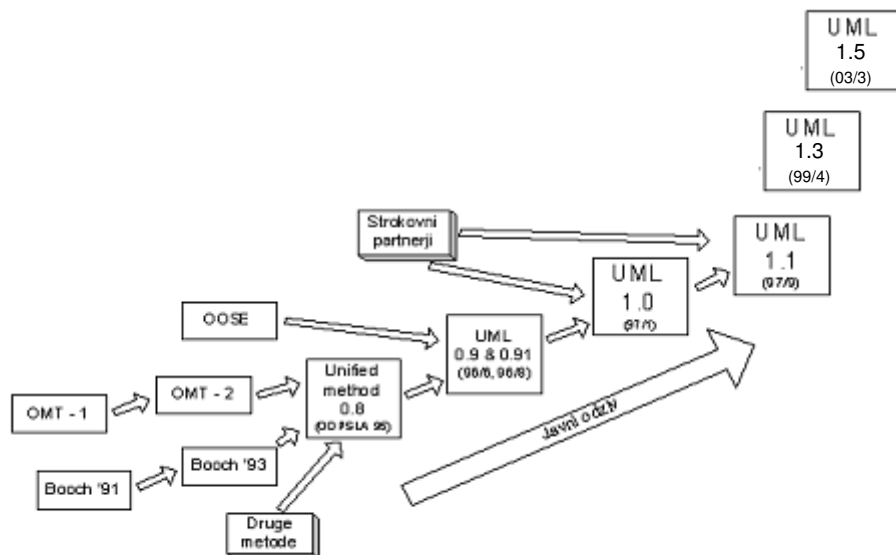
³⁶ Unified Modeling Language

podatkov in popolno podporo vsemu, kar jezik za objektno modeliranje ponuja. Ob standardizaciji notacije kot primarnem cilju so avtorji (Booch, Jacobson, Rumbaugh) poskušali vzporedno doseči tudi jasen pogled na proces razvoja programskih sistemov. UML je sicer mogoče aplicirati s poljubnim razvojnim procesom, saj zajema vse diagramске tehnike, kljub temu pa so avtorji UML razvili tudi proces razvoja, t.i. Poenoteni proces, o katerem bo govora v nadaljevanju in ki predstavlja podlogo za oblikovanje lastnega procesa razvoja.

Zgodovinsko gledano je UML nastal z združitvijo več metodologij za objektno orientirano analizo in načrtovanje, med katerimi so bile najpomembnejše:

- OMT avtorja James-a Rumbaugh-a,
- OOSE avtorja Ivar-ja Jacobson-a,
- Booch-93 metodologija avtorja Grady-a Booch-a.

Slika 19: Razvoj UML



Vir: Pomen jezika za objektno modeliranje UML, 2005.

UML je bil kasneje v okviru OMG (Object Management Group) standardiziran in je trenutno standardni jezik za modeliranje sistemov.

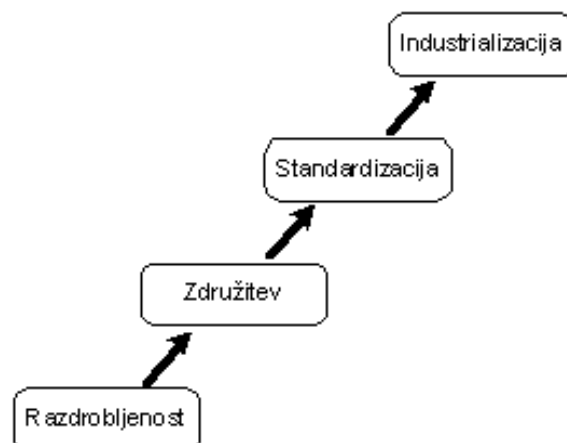
Medtem ko večina objektnih metodologij sestoji iz modelirnega jezika in procesa razvoja, UML podaja samo modelirni jezik (notacijo), ki jo metodologije uporabljajo za izražanje modelov. Zato UML imenujemo jezik za objektno modeliranje in ne metodologija. Za komunikacijo o rezultatih razvoja med razvijalci namreč potrebujemo enotno notacijo, ki prikaže rezultate dela in nas ne obremenjuje s postopki, ki so privedli do teh rezultatov. Zaradi velikega spektra različnih procesov razvoja, ki so posledica prilagajanja problemskim domenam, je poenotenje metodologij zelo zahtevno opravilo.

4.2.2.1 Cilji in namen aplikacije UML

Osnovne cilje razvoja jezika UML lahko strnemo v naslednjih točkah (Pomen jezika za objektno modeliranje UML, 2005):

- Razviti vizualni modelirni jezik, ki bo uporabnikom omogočal razvoj in izmenjavo modelov. UML naj se osredotoča na razmeroma majhno množico osnovnih konceptov, ki zadovoljujejo potrebe v razvoju večine programskih sistemov in ki jih ni nujno vedno uporabiti v polnem številu.
- Omogočiti razširjanje ali specializacijo osnovnih konceptov in s tem prilagajanje jezika posebnim potrebam projektov ne da bi predefinirali osnovne koncepte.
- Razviti jezik, ki bo neodvisen od programskega jezika in razvojnega procesa, kar zagotavlja podporo široki paleti projektov, kjer lahko jezik UML uporabimo brez modifikacij.
- Razviti formalno osnovo za razumevanje modelirnega jezika, ki naj pomaga uporabniku pri spoznavanju in uporabi jezika. UML naj definira formalno definicijo statičnega formata modela s pomočjo metamodela, predstavljenega v obliki razrednih diagramov.
- Vzpodbuditi rast trga objektnih orodij, saj poenotena razvojna notacija omogoča boljšo podporo s strani dobaviteljev in lažjo izmenjavo podatkov.
- Razviti podporo za visokonivojske razvojne koncepte kot so sodelovanja, ogrodja, vzorci in komponente. Jasno definirana semantika naprednih konceptov je zelo pomembna za popolno izkoriščanje objektnih tehnologij in ponovne uporabe.
- Integrirati najboljše izkušnje iz prakse. Ključno vodilo v ozadju razvoja jezika UML je združevanje najboljših izkušenj iz praktičnih primerov z upoštevanjem različnih pogledov na nivoje abstrakcije, arhitekture, problemska področja, faze razvoja, tehnike implementacije itd.

Slika 20: Proces uveljavljanja UML



Vir: Pomen jezika za objektno modeliranje UML, 2005.

Slika 20 prikazuje pomen naštetih točk. Jezik za objektno modeliranje UML je nastal iz potrebe po poenotenosti in združitvi objektnih razvojnih metod, ki obstajajo v svetu razvoja programskih sistemov. Standardizacija jezika omogoča odločitev za tehnologijo in ne za proizvajalca, kar od razvijalcev podpornih orodij zahteva določeno mero sodelovanja, če

želijo doseči medopravilnost orodij. Osnova za to je standarden jezik. Nenazadnje mora jezik biti uporabljen v industrijskem okolju, da bo doživel svoje razširjanje in možnost izkoriščanja. Delo na tem področju danes teče zelo aktivno in pričakujemo lahko, da bodo vsa podporna orodja v prihodnosti podpirala UML. Vodilna podjetja že danes s svojimi orodji podpirajo UML in omogočajo uporabnikom, da se odločijo za modelirni jezik kot tak in ne za njihovo »izpopolnjeno« implementacijo priporočil ali standardov.

4.2.2.2 Predstavitev jezika UML

UML definira notacijo, metamodel in razširitvene mehanizme jezika. Notacija je grafična predstavitev jezika (sintaksa), ki sama po sebi ne daje jasnega in nedvoumnega pomena posameznih konstruktov. Splošna uporaba notacije največkrat že definira nekatere neformalne definicije, toda bolje je uporabiti formalno definicijo. Ideja rigoroznih specifikacij in jezikov načrtovanja je prevladujoča na področju formalnih metod. Če uporabljamo tehnike s tega področja, izražamo specifikacije in načrtovanja s pomočjo izpeljank predikatnega računa. Takšne definicije so matematično natančne in ne dopuščajo dvoumnosti, a na žalost niso vsesplošno uporabne. Tako npr. lahko dokažemo, da program zadovoljuje matematični specifikaciji, nikakor ne moremo dokazati, da te specifikacije v resnici izpolnjujejo uporabnikova pričakovanja.

Večina objektnih metod ima v svoji definiciji le malo rigoroznosti, zato so avtorji jezika za objektno modeliranje UML poizkušali jeziku dati formalno podlago, ne da bi žrtvovali njegovo uporabnost. Ena od možnosti je definicija metamodela (modela o modelih). Največkrat gre pri tem za razredni diagram, ki definira notacijo in natančno ter nedvoumno definira pomen posameznih gradnikov. Avtorji so jezik za objektno modeliranje UML zgradili štiriplastno (OMG Unified Modeling Language Specification, 2005):

- *meta-metamodel*, ki definira jezik za specifikacijo metamodela in predstavlja infrastrukturo za arhitekturo metamodela,
- *metamodel*, ki je primerek meta-metamodela in definira jezik za specifikacijo modelov,
- *model*, ki je primerek metamodela in definira jezik za opis informacijske domene,
- *uporabniški objekti*, ki so primerki modela in definirajo informacijsko domeno.

Takšna zasnova jezika omogoča uporabo jezika na vseh področjih, saj vsebuje gradnike, ki omogočajo modeliranje različnih sistemov. Poleg formalne zasnove jezika so avtorji v jezik dodali tudi jezik OCL³⁷, ki določa omejitve in je popolnoma formalen ter omogoča oblikovanje omejitev vseh modelov (vloge v asociacijah, števnost povezav, pred- in po-pogoji metod ipd.). Kot kažejo izkušnje, je kombinacija formalne in neformalne zasnove jezika za modeliranje prinesla prednosti na dveh področjih. Po eni strani je jezik splošno uporaben, a jasno definiran, kar omogoča uporabo jezika UML na vseh problemskih področjih, po drugi strani pa jasno definirana struktura jezika omogoča lažji razvoj podporne programske opreme.

³⁷ Object Constraint Language

4.2.2.3 Gradniki UML

V nadaljevanju bo podana kratka teoretična osnova zgradbe UML ter semantičnih pravil notacije UML. Trenutno je namreč na razpolago številna literatura s tega področja, zato na tem mestu podajam le osnove. Sama aplikacija UML bo prikazana v naslednjem poglavju na primeru podatkov in procesa vodenja distribucijskega elektroenergetskega sistema.

K osnovnim gradnikom UML prištevamo (Booch, Rumbaugh, Jacobson, 1999, str. 18-26):

- elemente,
- povezave,
- diagrame.

Elementi so namenjeni opisovanju strukture, obnašanja, združevanja ali opomb. Omogočajo prikaz različnih pogledov na sistem.

Strukturni elementi, ki prikazujejo statični pogled na sistem, so (Rumbaugh, Jacobson, Booch, 1999, str. 23-35):

- Razred je opis niza objektov, ki imajo enake attribute, operacije, povezave in semantiko. Razred implementira enega ali več vmesnikov. Grafično je predstavljen kot pravokotnik, ki vsebuje vsaj ime razreda, njegove attribute in operacije.
- Vmesnik je zbirka operacij, ki opredeljuje storitve, ki jih ponujata razred ali komponenta. Vmesnik opredeljuje specifikacijo in parametre operacije, izvedba operacije je skrita. Grafično je prikazan kot krogec.
- Sodelovanje opredeljuje interakcijo in je združba vlog in ostalih elementov, ki skupno delujejo v kooperativnem obnašanju. Razred je lahko del več sodelovanj. Sodelovanje je grafično predstavljeno kot črtkasta elipsa.
- Primer uporabe je opis niza zaporednih korakov, ki jih izvaja sistem, in določenemu akterju prinašajo rezultat, ki ga lahko opazujemo. Grafično je prikazan kot elipsa.
- Komponenta je fizični del sistema, ki zagotavlja uresničitev niza vmesnikov. Komponenta tipično predstavlja fizično pakiranje logičnih elementov, kot so razredi, vmesniki in sodelovanja. Grafično je prikazana kot pravokotnik z zankama.
- Vozlišče je fizični element, ki obstaja v času izvajanja in predstavlja računalniški vir, ki ima lasten spomin in procesorsko moč. Grafično je prikazan kot kocka, ki vsebuje lastno ime.

Dinamični elementi, ki prikazujejo obnašanje sistema, so naslednji:

- Interakcija predstavlja obnašanje, ki vsebuje sporočila, ki se z določenim namenom izmenjujejo med objekti v določenem kontekstu komunikacije. Interakcija vsebuje različne elemente, in sicer sporočila, obnašanja, ki jih sporočila sprožijo, ter povezave med objekti. Grafično predstavljamo sporočilo kot puščico, ki vsebuje ime operacije, ki jo sproža.
- Diagrami stanj predstavljajo obnašanje, ki opredeljuje zaporedje stanj objekta, ki se menjajo v interakciji kot odgovor na dogodke. Vključuje stanja, prehode (tok iz stanja v stanje), dogodke (ki sprožajo prehode) in aktivnosti (odgovore na prehode). Grafično

stanje prikazujemo kot pravokotnik z zaobljenimi koti, ki vsebuje ime in vgnezdena podstanja, če le-ta obstajajo.

Mehanizem za združevanje objektov v UML se imenuje paket. Paket je splošni mehanizem združevanja elementov. Za razliko od komponente in vozlišča, ki obstajata v času izvajanja, paket obstaja na konceptualnem nivoju. Grafično je prikazan kot mapa.

Za dodajanje opomb in opisov v UML uporabljamo ustrezne elemente opomb, ki se imenujejo zapiski. Zapisek lahko pripnemo na element ali skupino elementov. Grafično je prikazan kot zavihan list.

Povezave predstavljajo odnose med razredi in so abstraktna predstavitev odnosov realnega sveta. Povezava med odvisnima razredoma je primer povezave med razredoma.

Povezave v UML lahko razdelimo na:

- Odvisnosti so semantične povezave med dvema elementoma, pri katerih sprememba neodvisnega elementa lahko povzroči spremembo odvisnega elementa. Grafično jih prikažemo kot črtkano puščico.
- Asociacije so strukturne povezave, ki opisujejo relacije, ki povezujejo objekte. Poseben primer asociacije je agregacija, ki je opisana v podpoglavju 4.2.1. Asociacijo prikažemo kot polno črto med dvema ali več strukturnimi elementi. V primeru agregacije je na strani črte, ki je pri elementu, ki predstavlja celoto, votel romb, v primeru agregacije kot primera kompozicije (sestavljene agregacije) pa poln romb. Za podrobnejši opis asociacije lahko na vsako stran črte dodamo ime vloge, saj imajo binarne asociacije natančno dve vlogi. Za imena vlog uporabljamo samostalnike. Z vlogo opišemo, kako razred vidi drugi razred preko asociacije. Uporaba vlog je nujna, če med istimi razredi obstaja več asociacij. Če je med razredoma le ena asociacija, vlog ne uporabljamo, razen če želimo asociacijo natančneje opisati. Binarne asociacije povezujejo dva razreda, n-kratne asociacije pa povezujejo več razredov. Posebna oblika asociacije je rekurzivna asociacija. Asociacija razreda ne kaže na drugi razred, temveč kaže na samo sebe. Uporaba vlog pri teh vrstah asociacij je nujna. Asociacijam lahko določimo tudi kardinalnost oz. števnost. Tako določimo, koliko objektov enega razreda je lahko povezanih z objekti drugega povezanega razreda. Tudi asociacije lahko vsebujejo attribute in operacije. Te asociacije imenujemo razredne asociacije. Razred asociacije je pri tem definiran kot ostali razredi. Zanj je značilno samo to, da sodeluje v asociaciji. Ti razredi se imenujejo asociacijski razredi ali povezovalni razredi.
- Generalizacije predstavljajo povezave generalizacije in specializacije, ki sta že opisani v podpoglavju 4.2.1. Grafično se prikažejo kot puščica z votlo konico, usmerjeno proti predniku oziroma nadrazredu.
- Realizacije so semantična povezava med elementoma, kjer prvi opredeljuje funkcijo, ki jo drugi izvede. Srečamo jo med vmesniki in razredi ter komponentami, ki jih uresničujejo, in med primeri uporabe ter sodelovanji, ki jih uresničujejo. Grafično so prikazane kot črtkana puščica z votlo konico.

Diagrami oziroma diagramske tehnike jezika UML omogočajo štiri različne poglede na problemsko področje. Prvi je pogled na uporabniške zahteve, kjer uporabljamo diagrame primerov uporabe, ki izvirajo neposredno iz Jacobsonove OOSE³⁸ metodologije. V drugo skupino lahko uvrstimo statični pogled na problemsko domeno, kjer uporabljamo razredne diagrame (nekoliko spremenjena diagramska tehnika povzeta iz Rumbaugh-ove OMT³⁹ metodologije) in objektne diagrame. Tretji pogled predstavlja skupina diagramskih tehnik, ki modelirajo obnašanje sistema in interakcijo med objekti sistema. Sem spadajo diagrami stanj, s katerimi opišemo prehajanje stanj za posamezne razrede, in diagrami aktivnosti, ki prikazujejo izvajanje aktivnosti sistema. Interakcijo med objekti, ki zagotavljajo funkcionalnost celotnega sistema, prikazujejo diagrami zaporedja, ki prikazujejo tipično interakcijo objektov v scenarijih uporabe sistema, in diagrami sodelovanja, ki prikazujejo način sodelovanja med množico objektov z namenom izvrševanja določene funkcionalnosti sistema. Četrty pogled na sistem omogoča skupina diagramov implementacije, s katerimi prikažemo implementacijske konstrukte. V jeziku UML so to diagrami komponent, ki prikazujejo organizacijo programskih komponent, in diagrami razvrstitve (porazdelitve)⁴⁰, s katerimi prikažemo konfiguracijo programskega sistema in strojnega okolja, kjer se sistem izvaja. Na tem mestu naj dodam, da obstaja nevarnost, da bi lahko veliko število diagramskih tehnik uporabnika zmedlo in ga usmerilo v uporabo tehnologije zaradi tehnologije same. Namen jezika UML je uporabiti le tiste diagramske tehnike, ki so pomembne za določeno problemsko področje, nikakor pa ni nujno uporabiti vseh tehnik na vseh področjih.

Poleg navedenih pogledov na sistem in njihovih gradnikov UML kot novost uvaja koncept stereotipov, s katerim lahko poljubnemu elementu jezika definiramo oziroma razširimo nov pomen oziroma pomen prilagodimo zahtevam razvojnega procesa, ter koncept vzorcev, ki jih jezik UML kot prvi uvaja v proces razvoja. Vzorec je v jeziku UML lahko uporabljen praktično kjerkoli in pomeni jedro idejne rešitve, ki je v preteklosti vodila do uspešne implementacije. Nov je tudi koncept razvoja na podlagi komponent, ki uvaja komponento kot gradnika programskih sistemov. Komponenta je definirana kot množica razredov, ki je ločena od sveta z jasno definiranim vmesnikom.

4.2.3 Poenoteni proces razvoja informacijskega sistema⁴¹

Kot omenjeno je jezik UML modelirni jezik in ne metodologija razvoja programske opreme in informacijskih sistemov, saj nima definiranega procesa razvoja, ki je temelj metodologije in poleg modelirnega jezika tudi njen sestavni del. Avtorji jezika UML so zato pripravili izdajo poenotenega procesa razvoja, ki naj bi združeval najboljše postopke obstoječih procesov. Avtorji so že na samem začetku ugotovili, da natančno definiranega procesa razvoja ne bi bilo mogoče razviti, zato so razvili ogrodje, kjer uporabnik lahko sam prilagodi ponujeno razvojno ogrodje svojim zahtevam in uporabi samo tehnike, ki so primerne in smiselne za konkretno modelirani proces.

³⁸ Object Oriented Software Engineering

³⁹ Object Modeling Technique

⁴⁰ Deployment Diagram

⁴¹ The Unified Process

Proces razvoja programske opreme predstavlja opredelitev aktivnosti, ki so potrebne za implementiranje in spreminjanje potreb uporabnikov v odgovarjajoče produkte, ki predstavljajo programsko rešitev, ter za spreminjanje dodatnih sprememb teh zahtev v nov, ponovno konsistentno odgovarjajoči niz produktov (Booch, Jacobson, Rumbaugh, 1999, str. 24). Ti produkti poleg delujoče programske opreme obsegajo še množico med seboj konsistentnih modelov, ki prikazujejo različne poglede na sistem in so dokumentirani s pomočjo UML jezika. Iterativno inkrementalna narava objektnega pristopa ima za posledico prekrivanje med posameznimi fazami razvoja, kar narekuje predstavitev procesnega modela v dveh dimenzijah: časovno po fazah razvoja in organizacijsko po aktivnostih, ki jih opravljamo znotraj vsake ponovitve. Po časovni komponenti so opredeljene faze razvoja, ki zajemajo celoten življenjski cikel razvoja informacijskega sistema (Arlow, Neustadt, 2002, str. 32):

- začetna faza⁴²,
- zbiranje informacij⁴³,
- konstrukcija⁴⁴,
- prevzem⁴⁵.

Tovrstna predstavitev v procesnem modelu poveže dva do sedaj ločena pogleda na razvoj programske opreme oziroma informacijskega sistema. Gledano s časovne perspektive govorimo o razvojnih ciklih, fazah, iteracijah in mejnikih, kar je značilno za upravljalni nivo gledanja na projekt. Tehnično gledano pa je proces razvoja sestavljen iz aktivnosti kot komponent delovnih tokov procesa razvoja.

Inkrementalnost procesa razvoja se kaže ob razvoju posameznih segmentov oziroma produktov znotraj posamezne faze razvoja, iterativnost pa v ponavljanju posameznih korakov, ki vodijo do zelenega cilja. Vsaka iteracija je sestavljena iz analize, načrtovanja, izvedbe in testiranja faznega produkta. Princip iterativnosti se v praktičnih primerih večinoma uporablja v fazi gradnje, poenoteni proces pa omogoča njegovo aplikacijo tudi v preostalih fazah razvoja.

Slika 21 prikazuje dvodimenzionalni pogled na razvojni proces. Na sliki lahko vidimo, kako so posamezne faze organizirane po času in si sledijo zaporedno, hkrati pa je potrebno gledati tudi organizacijo po aktivnostih in njihov delež, ki se uporablja v posamezni fazi. Vidimo, da se posamezne aktivnosti lahko uporabljajo v vseh oziroma vsaj v večjih fazah procesa razvoja.

Življenjski cikel vsake generacije programske opreme oziroma produktov posamezne faze razvoja torej razdelimo na navedene štiri faze. Vsaka faza se zaključi z definiranim časovnim mejnikom in zahteva sprejetje odločitev, ki lahko v veliki meri vplivajo na uspešnost projekta.

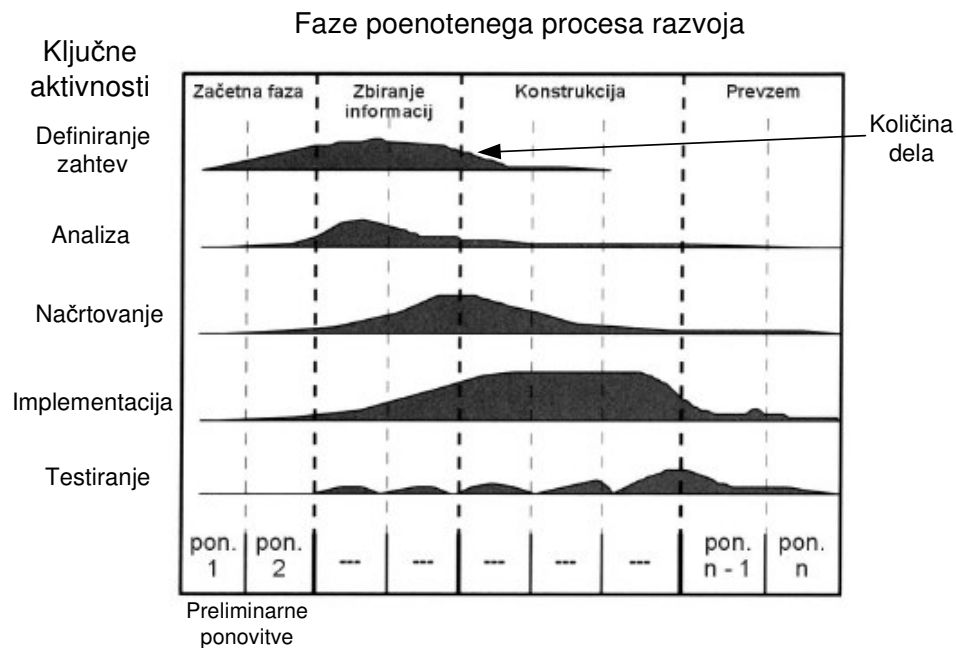
⁴² Inception

⁴³ Elaboration

⁴⁴ Construction

⁴⁵ Transition

Slika 21: Razčlenitev poenotenega procesa



Vir: Arlow, Neustadt, 2002, str. 32.

Bistveni cilji posamezne faze so sledeči (Arlow, Neustadt, 2002, str. 33-39):

- **Začetna faza** definira namen in cilj projekta, izvedljivost projekta glede na poslovne zahteve, definira projekt s poslovnega stališča, določi njegov obseg in dobrobiti izvedbe. Pod pojmom projekt je tu mišljen proces razvoja informacijskega sistema. Poleg tega ocenimo, kako obsežen bo projekt, kakšna so potrebna vlaganja, kakšni so potrebni viri in katera so kritična tveganja izvajanja projekta. Že v tej fazi zajamemo bistvene primere uporabe in se torej predvsem osredotočimo na izvajanje aktivnosti definiranja zahtev in analize. Če v to fazo vključimo tudi izdelavo prototipa za potrditev osnovnega koncepta, potem se v določeni meri izvajata tudi aktivnosti načrtovanja in implementacije prototipa. Prototip nima vpliva na končni produkt, zato je kot tak kasneje zavržen in zato tudi ni podvržen aktivnosti testiranja.
- **Faza zbiranja informacij** v grobem obsega analizo problemskega področja, postavitev ogrodja arhitekture, izdelavo projektnega plana za fazo konstrukcije, zbiranje začetnih zahtev za funkcionalnost sistema in izpopolnitev opredeljenih tveganj. V prvi fazi razpolagamo z okvirnimi idejami o zahtevah glede informacijskega sistema. Nato zberemo podrobne podatke o zahtevah, izvedemo analizo in načrt ter določimo osnovno arhitekturo novega sistema. Na koncu določimo plan konstrukcije. Na osnovi vseh teh podatkov opredelimo, kaj in kako bomo delali, ter definiramo tveganja:
 - **tveganje potreb**, ki vključuje nevarnost, da zgradimo informacijski sistem, ki ne odgovarja postavljenim zahtevam in je lahko posledica nesporazuma med razvijalcem in uporabnikom;
 - **tehnološko tveganje**, kjer se poraja vprašanje, ali imamo na razpolago ustrezno tehnologijo za rešitev problema, saj se namreč lahko dogodi, da uporabimo več

različnih tehnoloških rešitev, ki so dobre, integracija med njimi pa je težavna ali celo nemogoča. Poraja se tudi vprašanje o posledicah v primeru odpovedi sistema;

- tveganje kadrov, ki vključuje dilemo, ali imamo na voljo dovolj sodelavcev in ali so ustrezno usposobljeni za izvedbo projekta. Pri večini projektov se izkaže, da izobraževanje ni bilo izvedeno v zadostni meri, zato je potrebno izobraževanju in usposabljanju posvetiti posebno skrb;
- tveganje, ki vključuje (ne)zadostnost podpore in (ne)zaupanje vodstva glede realizacije projekta.

V tej fazi določimo do 80% primerov uporabe. Primeri uporabe tvorijo ogrodje informacijskega sistema.

- **Faza konstrukcije** vključuje planiranje izgradnje in samo izgradnjo. V tej fazi so primeri uporabe in diagrami razredov osnova za delo. Izgradnja je pretežno izvedena v okviru aktivnosti analiziranja, načrtovanja, izvedbe in testiranja. Korake izgradnje ponavljamo do doseženega ustreznega nivoja kakovosti, kar je uporabniško testirano in potrjeno. Ta način se uporablja zaradi zmanjšanja tveganja. Faza vključuje več ponavljanj, znotraj katerih postopno dograjujemo sistem. V vsaki ponovitvi se dodaja funkcionalnost primerom uporabe, izvedena sta testiranje in integracija. Uporabniki sodelujejo pri testiranju vsake ponovitve in lahko vplivajo še na zadnje korekcije izdelanih funkcionalnosti. Ob izdelani programski opremi pripravimo tudi ustrezno tehnično in uporabniško dokumentacijo, ki vključuje tudi UML diagrame. Dopolnimo tudi vse nedokončane modele iz prejšnjih faz. Na koncu faze je potrebno oceniti, ali je programska oprema primerna za vsakodnevno uporabo in predajo uporabnikom.
- **Faza prevzema** vključuje predajo izdelka v uporabnikovo okolje. Zadnja faza je faza prehoda. V tej fazi izvedemo zadnja testiranja, merimo zmogljivosti sistema in izobrazimo uporabnike. Po zagonu se lahko pokažejo težave, ki jih v fazi testiranja nismo uspeli identificirati. V kolikor so odstopanja večja, lahko odpravo problema posredujemo v naslednjo ponovitev izdelave sistema. Manjše popravke izvedemo kar znotraj faze prevzema. Tako dopolnimo model implementacije in testiranja ter zabeležimo pridobljene izkušnje za izboljšanje procesa v naslednjih ponovitvah.

Razvoj programske opreme na opisan način omogoča kontinuirano dopolnjevanje in spreminjanje funkcionalnosti glede na zahteve poslovnega okolja in uporabniške zahteve. Ključnega pomena je postopnost in iterativnost ter možna povezanost posameznih modelov. To omogoča enostavno odkrivanje potrebnih sprememb v modelih, ki so bližji implementaciji, glede na spremembe v modelu primerov uporabe.

Drugo dimenzijo poenotenega procesa predstavljajo statične komponente, ki so vključene v posamezne faze procesa in vsebujejo aktivnosti:

- **zajemanje zahtev** - definiranje funkcionalnosti sistema preko pridobivanja uporabniških zahtev,
- **analiza** - natančnejše opredeljevanje in dograjevanje ter strukturiranje funkcionalnosti,

- **načrtovanje** - realizacija funkcionalnosti v okviru definiranja logične arhitekture sistema,
- **implementacija** - izgradnja programske kode,
- **testiranje** - izvajanje integracijskega preverjanja implementacije sistema

Zajemanje zahtev

Namen je preko primerov uporabe opredeliti model funkcionalnosti bodočega sistema. Zahtevano funkcionalnost opredelimo, ko opišemo načine uporabe sistema in rezultate, ki jih sistem nudi uporabnikom. Pomembno je identificirati vloge, ki komunicirajo s sistemom, s čimer so opredeljeni akterji sistema. Primere uporabe zberemo v model s pomočjo diagramov primerov uporabe. Poleg primerov uporabe, ki so glavno orodje za zajemanje zahtev, si pomagamo še z diagrami za opisovanje obnašanja sistema (npr. z diagrami zaporedja). Poleg tega lahko naredimo tudi nekatere prototipe uporabniških vmesnikov, ki pomagajo k podrobnemu razumevanju načina interakcije uporabnika in sistema. Za razumevanje problemske domene lahko modelu primerov uporabe dodamo tekstovni opis z opisom akterjev in njihove vloge ter izvajanje primerov uporabe. Za razumevanje lahko dodamo tudi definicijo uporabljenih izrazov. Primerom uporabe opredelimo pomembnost in jih razvrstimo v vrstni red za nadaljnji razvoj.

Analiza

Analiza predstavlja pogled na način realizacije zahtevane funkcionalnosti, kjer je potrebno podrobno analizirati uporabniške zahteve, opredeljene v predhodnem koraku. Model analize je prikazan zgolj na konceptualnem nivoju, podrobnosti implementacije niso prikazane. Razlikuje se od modela primerov uporabe, saj je opisan v formalnem jeziku. Namenjen je razvijalcu sistema in prikazuje notranji pogled na sistem, ki je sestavljen iz razredov in paketov. Prikazuje način izvedbe funkcionalnosti znotraj sistema, ki je osnova za načrtovanje in opredeljuje način realizacije primerov uporabe s strani sistema. Opredelijo se razredi in njihove odgovornosti, ki jih zagotavljajo razredne operacije in vmesniki, ter paketi, ki bodo zagotavljali posamezne funkcionalnosti. Opredelijo se tudi povezave med razredi ter njihovo združevanje v pakete. Na miselnem nivoju se preizkusi delovanje primerov uporabe skozi tako opredeljeni model. Opredeli se tudi povezava med primeri uporabe in elementi modela analize, ki omogočajo njihovo izvajanje. Za opisovanje modela analize se uporabita predvsem razredni diagram in diagram sodelovanja. Odvisnosti med paketi, ki realizirajo posamezne primere uporabe, prikažemo v razrednem diagramu na višjem nivoju abstrakcije.

Načrtovanje

Znotraj aktivnosti načrtovanja opredelimo fizični model sistema, ki bo osnova za implementacijo. Opredelimo model načrta in model porazdelitve. V modelu načrta podrobno opredelimo razrede, povezave med njimi in dinamično obnašanje sistema. Dokončno opredelimo attribute razredov, opredelimo vidnost⁴⁶ (javna, zasebna, zaščitena) in opišemo

⁴⁶ visibility (public, private, protected)

metode. Metode lahko opišemo v naravnem jeziku ali pseudo kodi ter prepustimo programiranje naslednjemu koraku, lahko pa jih že programiramo v izbranem programskem jeziku. Povezave med objekti dokončno opredelimo glede števnosti, smeri navigacije in imenovanja vlog glede na predvideno okolje aplikacije. Prav tako je treba upoštevati predvideno tehnologijo shranjevanja podatkov, ko načrtujemo razrede, ki izvajajo zapisovanje podatkov v okviru transakcije. Pakete preoblikujemo v podsisteme. Treba je zagotoviti čim manjšo odvisnost med podsistemi, opredeliti ustrezne vmesnike in zagotoviti, da podsistem ponuja pravilno realizacijo operacij, ki so opredeljene z njegovimi vmesniki. Dinamično obnašanje sistema podrobno opišemo z diagrami zaporedja, diagrami stanj in diagrami aktivnosti. Tako je model načrta podrobna predstavitev sistema z upoštevanimi lastnostmi okolja, v katerem bo uresničen. Model porazdelitve vsebuje vozlišča, njihove lastnosti in povezave ter začetno razporejanje razredov in podsistemov v vozlišča. Za opisovanje modela uporabimo diagram porazdelitve.

Implementacija

Glavni rezultat te aktivnosti je izvorna koda, ki uresničuje izdelke predhodnih korakov in omogoča delovanje sistema. Implementacija vključuje posamezne programske komponente, ki implementirajo razrede in podsisteme, ki so bili opredeljeni v načrtovanju. Razrede implementiramo kot posamezne komponente, ki vsebujejo izvorno kodo. Implementacija poteka v več iteracijah in lahko tudi vzporedno, zato je pomembno načrtovati integracijo po vsaki iteraciji. V okviru iteracij se izdelajo obvladljivi deli programske kode, ki se sproti integrirajo z že razvitim delom sistema. V tem koraku se izvede testiranje posameznih komponent, ki se nato integrirajo in namestijo na ustrezno strojno opremo. Celovito sistemsko preverjanje se izvede v koraku testiranja. V modelu uresničitve uporabljamo diagrama komponent in porazdelitve.

Testiranje

Namen testiranja je preizkus delovanja sistema, tako glede izpolnjevanja zahtevanih funkcionalnosti v smislu pravilnosti delovanja kot tudi izpolnjevanja nefunkcionalnih zahtev npr. performans. Testiranje na nivoju komponent je opravljeno v koraku implementacije, v koraku testiranja pa se izvaja, kot omenjeno, integracijsko in sistemsko testiranje. Za sistematičen način testiranja je pripravljen model testiranja, ki vsebuje (Booch, Jacobson, Rumbaugh, 1999, str. 313) testne primere, ki opredeljujejo vsebino testiranja, testne postopke, ki opredeljujejo način testiranja in testne komponente, ki omogočajo avtomatizacijo testnih postopkov. Plan testiranja mora vključevati urnik testiranja in osebe, ki bodo testiranje izvajale. Po izvedenem testiranju ocenimo uspešnost testiranja. V primeru uspešnega testa celotnega sistema je sistem pripravljen za predajo. Model testiranja dopolnjujemo in ga uporabimo ob vsaki novi ponovitvi znotraj življenjske dobe sistema.

V nadaljevanju bosta strnjeno prikazani še dve metodologiji razvoja informacijskih sistemov, ki sta namenjeni predvsem razvoju sistemov delujočih v realnem času in temeljita na predpostavkah Poenotenega procesa.

4.2.3.1 Metodologija ROPES⁴⁷

ROPES proces temelji na iterativnem življenjskem ciklu razvoja programske opreme in uporablja standardni UML metamodel za njegovo semantično ogrodje in notacijo (Douglass, 1999, str. 14). V ROPES so vključeni principi kot npr. iterativni in modelno-orientirani razvoj, dvosmerna asociativnost modeliranje-kodiranje, izgradnja zgodnjih izvršljivih modelov, preverjanje abstraktnih modelov znotraj načrtovanja itd., ki omogočajo razvoj še tako zahtevnih in kompleksnih sistemov delujočih v realnem času.

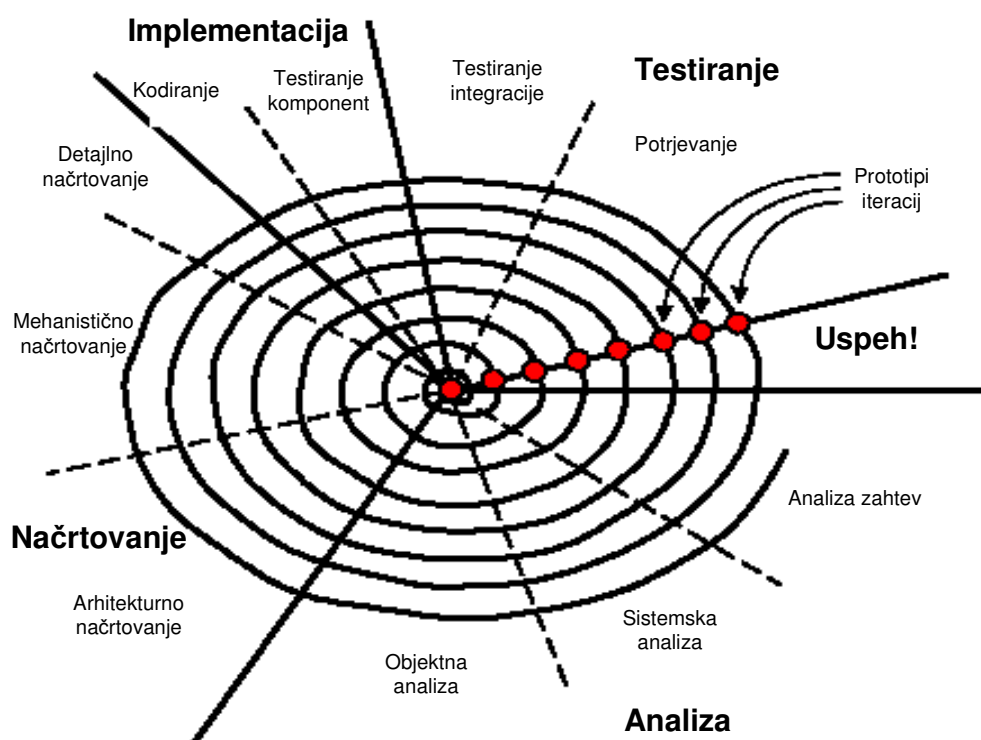
ROPES uporablja sledeče glavne aktivnosti v procesu razvoja (Douglass, 2000, str. 12-16):

- **Analiza**, ki definira bistvene aplikativne lastnosti sistema. Sestoji iz podsklopov:
 - Analiza zahtev detajlno identificira funkcionalne in performančne zahteve sistema in se ne spušča v notranjo strukturo sistema (načrtovanje).
 - Sistemska analiza se osredotoča na tri aktivnosti: definiranje optimalne dekompozicije na strojno in programsko opremo, izdelavo višje nivojske arhitekture sistema in na natančnejše definiranje kompleksnih kontrolnih algoritmov.
 - Objektna analiza sestoji iz dveh podfaz, in sicer analize objektne strukture in analize objektnega obnašanja.
- **Načrtovanje** določi partikularno rešitev, ki optimira aplikacije v skladu s projektnimi cilji, hkrati pa ostaja konsistentna z modelom analize. Sestoji iz treh podsklopov:
 - Arhitekturno načrtovanje identificira strateške odločitve načrtovanja, ki vplivajo na celotno aplikacijo, vključujoč povezavo na fizični razvrstitveni model, identifikacijo izvajalnih komponent in sočasnostni model.
 - Mehanistično načrtovanje dodaja objektom v sodelovanju optimalnejše obnašanje glede na nekatere sistemske optimizacijske kriterije.
 - Podrobno načrtovanje dodaja nizko nivojske informacijske parametre za optimiranje končnega delovanja sistema.
- **Translacija** kot predstavitev ali prenos sistema v fizično okolje pomeni izdelavo izvršljive aplikacije iz izdelanega načrtovalnega modela. Ne pomeni samo izdelave izvršljive kode, ampak tudi testiranje posameznih komponent translacije.
- **Testiranje** omogoča integralno preverjanje delovanja izvršilnih aplikacij glede na določene pravilnostne kriterije, da bi se tako identificirale pomanjkljivosti ali napake oziroma potrdila vsaj minimalna sprejemljivost aplikacije.

Potek izvajanja navedenih aktivnosti v ROPES iterativnem razvojnem procesu prikazuje slika 22.

⁴⁷ Rapid Object-Oriented Process for Embedded Systems

Slika 22: ROPES razvojni proces



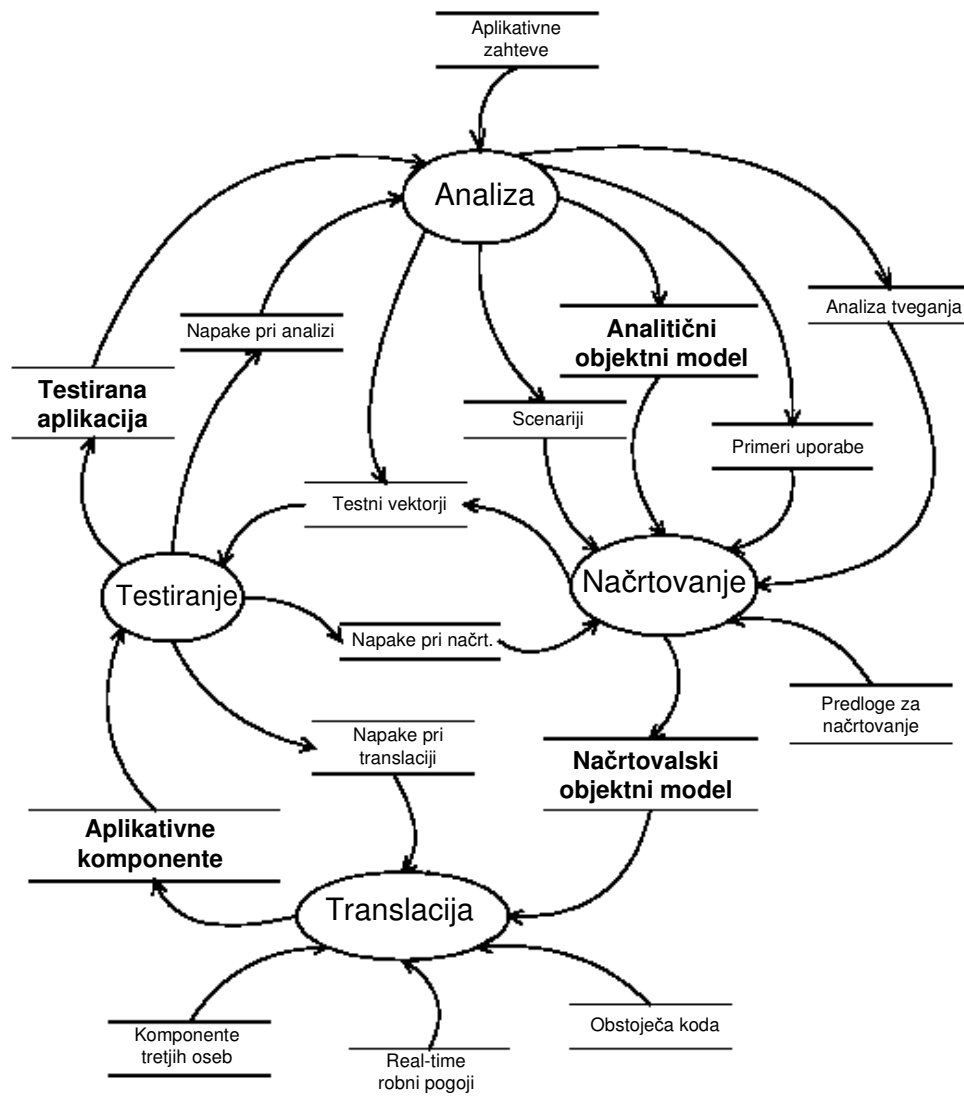
Vir: Douglass, 2000, str. 14.

ROPES proces v vsakem iterativnem ciklu proizvede delovni izdelek⁴⁸ (slika 23). Vsak iterativni cikel obsega aktivnosti, ki se izvajajo vedno na nekem nivoju abstrakcije in kot rezultat redosleda izvajanja teh aktivnosti so proizvedeni tudi delovni produkti posameznih iterativnih ciklov.

V praksi so produkti posameznih iteracij razvojni prototipi, ki predstavljajo vmesno verzijo systemskega modela ob koncu vsake faze razvoja. Tako prototipi sestojijo ne samo iz izvršilnih produktov, ampak tudi iz produktov, iz katerih so se ti izvršilni deli generirali (npr. urejeni primeri uporabe).

⁴⁸ artifact

Slika 23: ROPES delovni izdelki iteracij



Vir: Douglass, 1999, str. 16.

4.2.3.2 Metodologija COMET⁴⁹

COMET (Gomaa, 2000, str. 109-115) je namenska metodologija razvoja sočasnih⁵⁰ aplikacij v kontekstu življenjskega cikla razvoja aplikacije, posebej na področju distribuiranih aplikacij v realnem času. Kompatibilna je s koncepti Poenotenega procesa (npr. s konceptom inkrementalnosti in ponovljivosti ciklov). Predvsem se močno naslanja na koncept modeliranja sistemov s primeri uporabe⁵¹, ki določajo zaporedje interakcij med akterji in obravnavanim sistemom in so uporabljeni v več aktivnostih razvoja (zajemanje zahtev, analiza, načrtovanje). Na posamezen primer uporabe je namreč mogoče gledati različno detajlno. Pri modeliranju zahtev se tako definirajo funkcionalne zahteve v smislu identifikacije

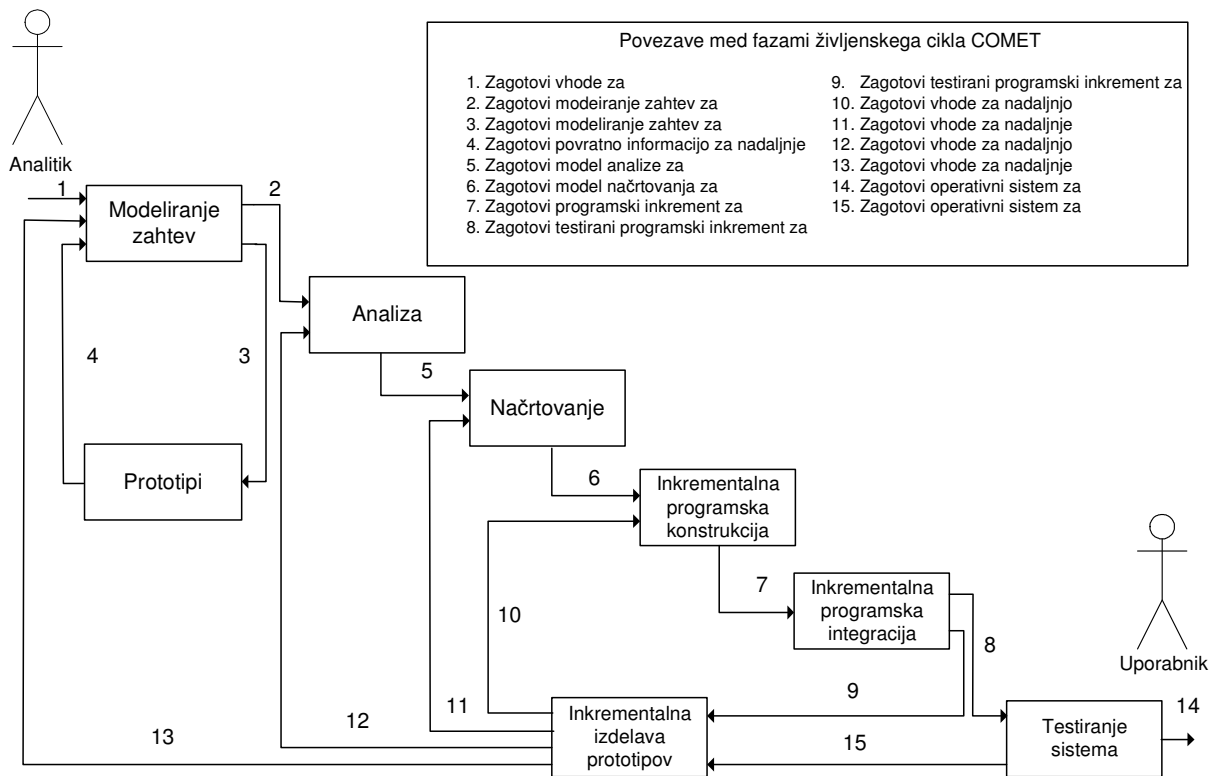
⁴⁹ Concurrent Object Modeling and architectural design mEThod.

50 Concurrent

51 Use-Case Modeling

akterjev in primerov uporabe funkcionalnosti, v modelu analize se primeri uporabe natančneje opredelijo glede opisa objektov, ki sodelujejo v primerih uporabe, in glede njihovih medsebojnih interakcij. V modelu načrtovanja je razvita arhitektura sistema vključujoč koncepte distribuiranosti, sočasnosti in skrivanja podatkov⁵². Slika 24 prikazuje celoten objektno-orientirani COMET model razvoja programske opreme.

Slika 24: COMET življenjski cikel razvoja



Vir: Gomaa, 2000, str. 110.

➤ Modeliranje zahtev

V tej aktivnostni fazi se definirajo osnovne funkcionalne zahteve obravnavanega sistema v smislu identifikacije akterjev, ki sodelujejo v posameznih primerih uporabe funkcionalnosti. Vsak primer uporabe je tudi besedno opisan. Pri tem je ključnega pomena sodelovanje uporabnikov in analitikov. Če zahtev ni mogoče jasno artikulirati, se kot pripomoček za opredelitev funkcionalnosti razvijejo prototipi, ki se jih potem zavrže.

➤ Model analize

Razvijejo se statični in dinamični modeli sistema. Statični model definira strukturne povezave med razredi problemske domene in je predstavljen z razrednimi diagrami. Na tej osnovi je nato razvit dinamični model, ki temelji na primerih uporabe iz prejšnje aktivnostne faze. Ti so natančneje definirani tako, da določijo objekte, ki sodelujejo v primerih uporabe, in njihovo medsebojno interakcijo. Interakcija je prikazana na diagramih sodelovanja ali zaporednostnih

⁵² Information Hiding

diagramih. Dinamični model tudi vključuje diagrame stanj za objekte, ki so lahko v različnih stanjih.

➤ **Model načrtovanja**

V tej fazi je izdelan načrt programske arhitekture sistema, kjer so produkti iz modela analize povezani z dejanskim operativnim okoljem izvajanja. To pomeni, da je problemska domena iz prejšnje faze nadgrajena z operativno rešitveno domeno, ki je produkt načrtovanja. Sistem je strukturiran v podsisteme, ki jih opredelimo kot agregatne objekte. Posebna pozornost je že v tej fazi namenjena načrtovanju distribuiranosti sistema v podsisteme, ki kot komponente komunicirajo med seboj z izmenjavo sporočil. Nato se pristopi k dejanskemu načrtovanju podsistemov. Predvsem za sekvenčne sisteme velja poudarek na objektno-orientiranih konceptih skrivanja podatkov, razredov in dedovanja. Pri načrtovanju sočasnostnih sistemov v realnem času z distribuiranimi tehnologijami odjemalec/strežnik je nujno upoštevati koncepte sočasne opravljalnosti⁵³.

➤ **Inkrementalna programska konstrukcija**

Aktivnosti te faze vključujejo izbor podsklopov sistema, ki bodo izdelani v posameznem inkrementu razvoja. Podsklopi so definirani z izborom primerov uporabe vključenih v predmetni inkrement razvoja in z odgovarjajočim izborom objektov. Aktivnosti vključujejo natančno načrtovanje, kodiranje in testiranje posameznih enot (razredov) programskih podsklopov. Programska koda se tako po inkrementalnih korakih postopno razvija v izdelani integralni sistem.

➤ **Integralna programska integracija**

Aktivnosti obsegajo integralno testiranje posameznega programskega inkrementa (podsklopa). Testiranje temelji na primerih uporabe, ki jih posamezni podsklop implementira. Za vsak primer uporabe je tako definiran testni primer integracije, ki vključuje predvsem testiranje vmesnikov implementiranih v objektih, ki vzpostavljajo povezave med objekti znotraj primerov uporabe. Vsak programski inkrementalni podsklop tvori t.i. inkrementalni prototip. Ko je programski inkrement ocenjen kot zadovoljiv, se pristopi h konstrukciji naslednjega inkrementa, ponovno preko iteracij znotraj faz inkrementalne programske konstrukcije in integracije. V primeru ugotovljenih večjih težav pri testiranju programskih inkrementov se ni mogoče izogniti ponovnim iteracijam v fazah modeliranja zahtev, analize in načrtovanja.

➤ **Testiranje sistema**

Zadnja aktivnostna faza vključuje funkcionalno testiranje sistema, to pomeni testiranje sistema glede na izražene in pričakovane funkcionalne zahteve. Tak način testiranja se, v nasprotju s tistim iz prejšnje faze, v žargonu imenuje testiranje »črne škatle«, saj temelji samo na funkcionalnosti primerov uporabe. Zato so tudi funkcionalni testni primeri izdelani za vsak primer uporabe znotraj »črne škatle«. Vsak izdelani inkrementalni podsklop mora skozi to fazo in skozi »roke« uporabnika.

⁵³ Concurrent Tasking

5 INFORMACIJSKI MODEL OPERATIVNEGA PROCESA V CENTRU VODENJA DEES

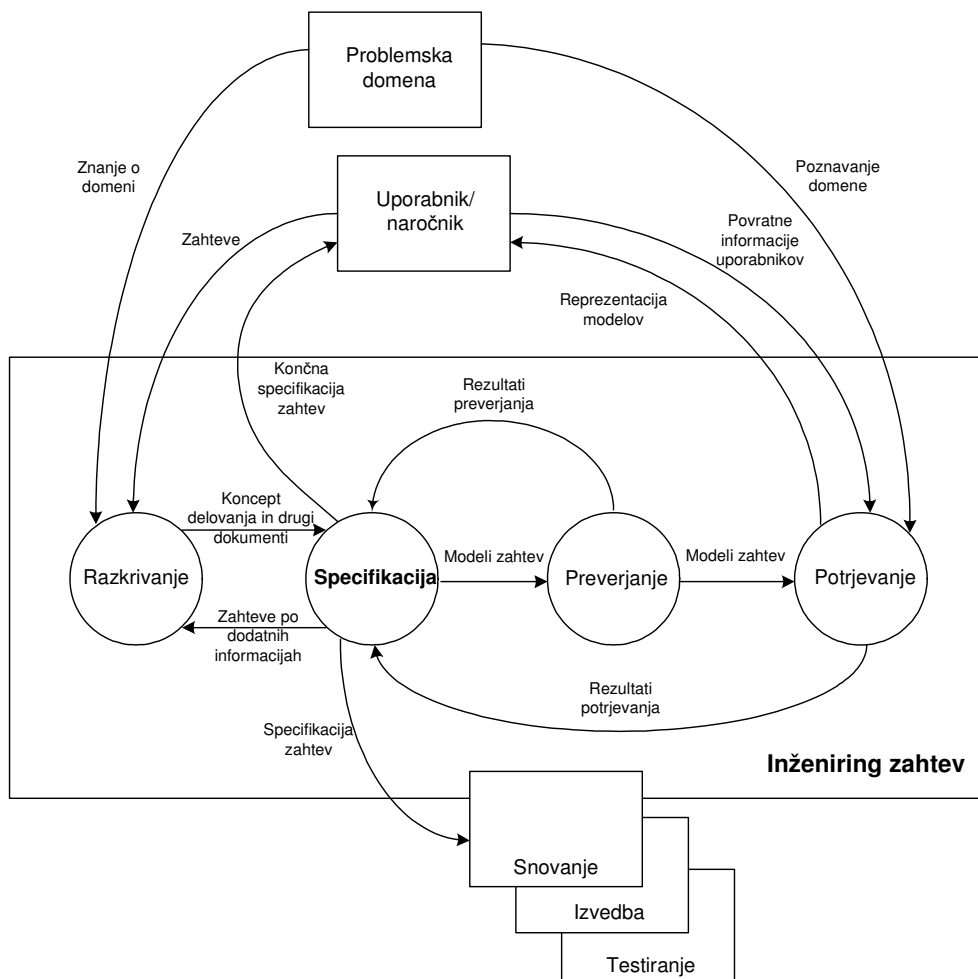
5.1 Nekaj o inženiringu zahtev za informacijski model sistema

Kot inženiring zahtev lahko opredelimo štiri najpomembnejše aktivnosti (Lipovec, 1998, str. 10):

- **Razkrivanje zahtev**, ki je sestavljeno iz zbiranja informacij, intervjujev, opazovanj in priprave nestrukturiranih dokumentov.
- **Specifikacija zahtev**, ki se zrcali v oblikovanju dokumenta z istim imenom in je uspešna, če kvalitetno zajame rezultate analize problema in je uporabna pri nadaljnjem delu.
- **Preverjanje in potrjevanje zahtev**, ki zagotovita kar se da pravilne in kvalitetne specifikacije.

Slika 25 prikazuje medsebojno povezavo aktivnosti in njihovo umestitev v življenjski cikel informacijskega sistema.

Slika 25: Umestitev inženiringa zahtev v razvoj informacijskega sistema



Vir: Lipovec, 1998, str. 10.

Tako kot je gradnja informacijskega sistema iterativen proces je tudi inženiring zahtev kot del procesa gradnje informacijskega sistema iterativen. V sklopu inženiringa zahtev je predvsem specifikacija zahtev ključna aktivnost, ki je osnova za dogovarjanje med naročnikom, uporabniki in razvijalci informacijskega sistema ter mora kot taka vsebovati načine sporazumevanja, ki so vsem udeležencem razumljivi. Ključna oseba, ki povezuje udeležence, je analitik. Ta je odgovoren za izdelavo vsaj dveh modelov sistemskih zahtev, in sicer uporabniško in razvojno usmerjenih. Modeli morajo biti izdelani kot modeli zahtev in njihove interpretacije za uporabnike.

V nadaljevanju bo predstavljena struktura teoretične specifikacije zahtev (Lipovec, 1998, str. 26-30), ki je ključna za konkretno modeliranje procesnih podatkov in procesov za potrebe operativnega vodenja DEES v luči funkcionalnosti sistema.

Znotraj specifikacije zahtev opredelimo **organizacijske zahteve**, ki omejuje domeno preučevanja, obravnavajo organizacijsko strukturo, cilje, dejavnike uspeha, poslovne aktivnosti ter udeležence v procesu in njihove odnose. Nadalje so **nefunkcionalne zahteve** predvsem tiste, ki opredeljujejo omejitve sistema. So statične in dinamične zahteve po učinkovitosti sistema in njegovih interakcij z uporabnikom, zahteve do zunanjih povezav z uporabniki, programsko, strojno in komunikacijsko opremo, zahteve določene v notranjih ali zunanjih standardih ter zahteve v zvezi v vodenjem kvalitete (sledljivost sprememb, varnost, vzdrževanje, razpoložljivost sistema). Med **funkcionalne zahteve** prištevamo tiste, ki opisujejo, kaj in kako naj bi sistem počel. Modeliramo jih z različnimi konceptualnimi modeli, ki odražajo različne poglede (vidike) na sistem.

Strukturni vidik na sistem se ukvarja z modeliranjem informacijske strukture domenskega prostora skupaj z omejitvami in dovoljenimi stanji teh struktur. Strukturni pogled na konkreten sistem v domenskem prostoru distribucijskega elektroenergetskega sistema bo obravnavan v podpoglavju 5.2.

Funkcionalni vidik prikazuje sistem kot mrežo procesov, kjer se vhodni podatki preslikujejo v izhodne. Procesi komunicirajo med sabo in z zunanjimi akterji (entitetami) s podatkovnimi tokovi. Funkcionalni pogled bo obravnavan na primerih v podpoglavju 5.3.

Vidik obnašanja, ki bo obravnavan v podpoglavjih 5.2 in 5.3, prikazuje sistem s koncepti stanj sistema, prehodi stanj in dogodki, ki sprožajo prehode stanj.

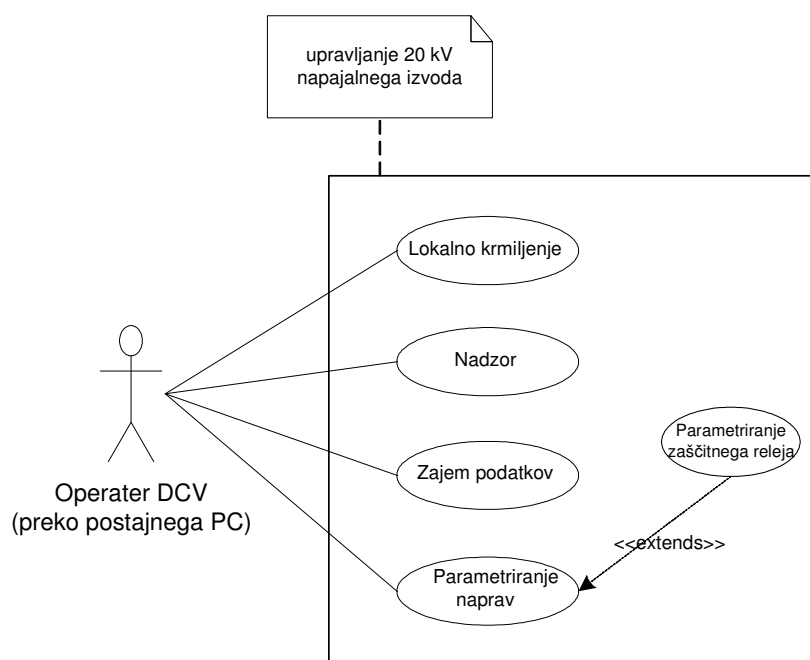
Vsi trije pogledi so med sabo povezani tako, da dogodki sprožajo aktivnosti (funkcionalni vidik), ki spreminjajo podatke (strukturni vidik) in s tem stanje sistema, ki določa možne nadaljnje dogodke (vidik obnašanja).

Tak način obravnave je zaobjet v objektnem vidiku, ki ima sicer izvor v strukturnem pogledu, je pa nadgrajen tako, da na problemsko domeno gleda skozi množico objektov, ki v sebi združujejo vse tri osnovne vidike. Razlika je v tem, da strukturni vidik modelira samo podatke, objektni vidik pa modelira obnašanje med sabo komunicirajočih objektov, kar narekuje tudi strukturo podatkov, ograjenih v teh objektih. V bistvu gre za aplikacijo objektne paradigme. Podpoglavji 5.2 in 5.3 dejansko opisujeta enoten objektni pogled na sistem, ločeni sta pa zato, ker za prikaz aplikacije modeliranja uporabljata različni poddomeni obravnavanega sistema.

Na slikah 26 in 27 sta prikazana dva načina specifikacije zahtev po UML notaciji, ki ju najdemo v literaturi, in sicer preko diagrama primerov uporabe (Douglass, 2003, str. 3) in zaporednostnega diagrama (Quatrani, 2001, str. 9). Predvsem zaporednostni diagram se, v

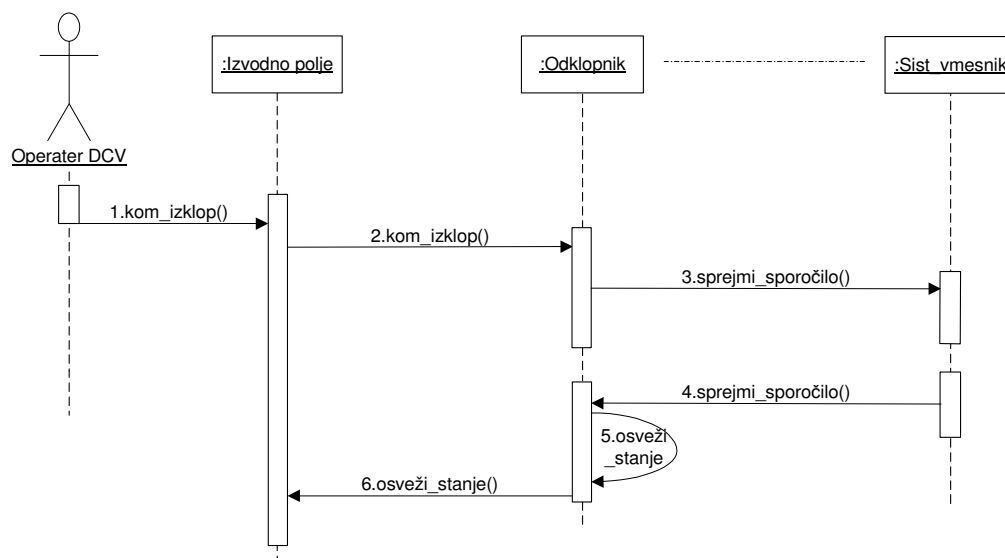
povezavi z diagramom sodelovanja, uporablja tudi za prikaz dinamičnega obnašanja sistema.

Slika 26: Specifikacija zahtev s primerom uporabe



Vir: Randić, 2000, str. 997.

Slika 27: Specifikacija zahtev z zaporednostnim diagramom



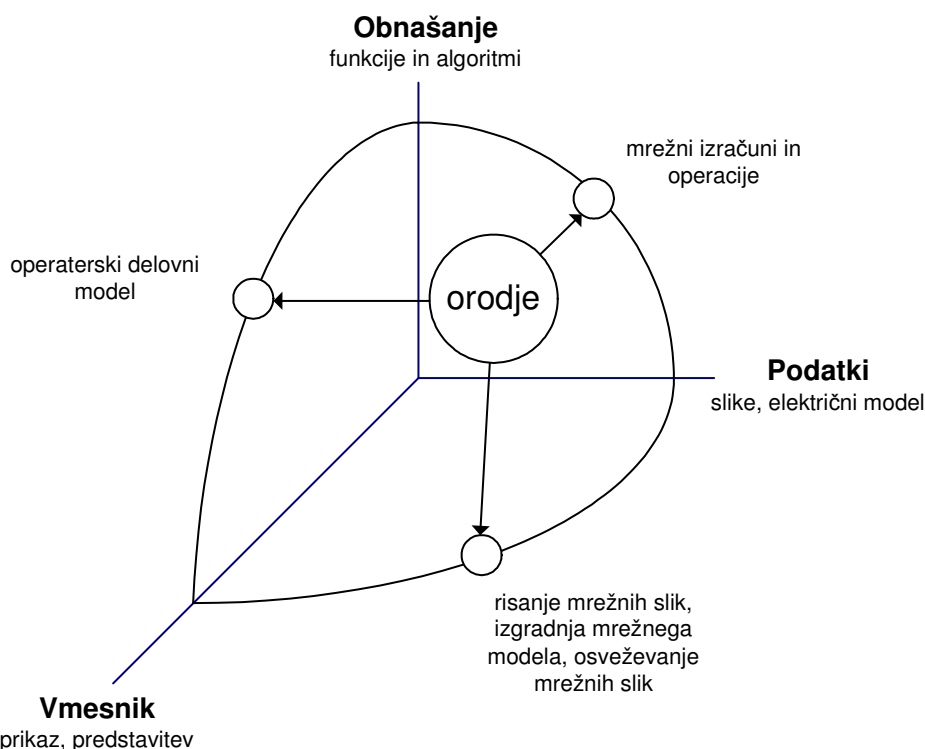
Vir: Dobraš, 2000, str. 122.

5.2 Objektno modeliranje procesnih podatkov

Objektno-orientirano modeliranje procesnih podatkov, elementov in nasploh elektroenergetskega omrežja temelji na konkretnih fizičnih napravah DEES, ki so preslikane v model objektov in njihovih tipov - razredov. Model objektov kot primerkov razredov vsebuje tri komponente (slika 28), kjer ima lahko katerakoli od njih prevladujočo vlogo:

- **Podatke**
- **Operacije** (obnašanje, kontrola, nadzor)
- **Grafični uporabniški vmesnik**

Slika 28: Komponente objektnega modeliranja



Vir: Hebel, 1997, 47.

Omeniti velja, da je potrebno izraz *model objektov* razumeti tudi širše. Objekt oziroma njegov tip je namreč lahko neposredna odslikava konkretne fizične narave realnega sveta, lahko pa je objekt tudi skupek realnih odslikav fizičnih naprav, in sicer kot agregacijski tip in hkrati kot matematični pripomoček (model), ki temeljne objekte poveže v odnos, iz katerega izhajajo fizikalne zakonitosti obnašanja. Primer takega objekta je matematični model elektroenergetskega omrežja, katerega gradniki so temeljni objekti fizičnih naprav in na katerem se izvajajo omrežni in topološki izračuni. V tem smislu je potrebno obravnavati tudi navedene tri komponente objektov (podatke, operacije, vmesnike), ki predstavljajo elektroenergetski sistem in ki so implementirane v podatkovni bazi omrežja (Handschin, 1997, str. 167).

V tem poglavju bo prikazano modeliranje procesnih podatkov elementarnih naprav⁵⁴ (aparatur) elektroenergetskih postrojev (EEP) kot gradnikov DEES, ki ga bom poimenoval Aparaturno-orientirano modeliranje procesnih podatkov, v nasprotju z v podpoglavju 3.4.2.1 opisanim Funkcijsko-orientiranim modeliranjem. Aparaturno-orientirano modeliranje bo ilustrativno preslikano v objektno-orientirane koncepte s pomočjo UML notacije.

5.2.1 Aparaturno-orientirano modeliranje

Aparaturno-orientirano modeliranje procesnih podatkov DEES temelji na konceptu naprave kot temeljnega podatkovnega izvora in temeljne organizacijske enote širšega elektroenergetskega postroja. Literatura (Šimunić, 2001, str. 34-42) navaja tri faze v postopku modeliranja:

1. *Strukturna sistemizacija signalov glede na tip naprave*, ki vključuje definiranje signalov, ki jih naprav generira ali sprejme,
2. *Analiza in dekompozicija elektroenergetskega postroja na tehnološke module*, ki vključuje povezavo med temi moduli in napravami,
3. *Definiranje funkcij in lokacij, kjer se posamezni signali uporabljajo*.

5.2.1.1 Strukturna sistemizacija signalov

Strukturna sistemizacija signalov predstavlja jedro modela elektroenergetskega postroja in obsega:

- identifikacijo različnih tipov naprav,
- informacijsko analizo tipov naprav (spremenljivke stanja in njihove domene vrednosti),
- definiranje strukture signalov.

Identifikacija tipov naprav izlušči vse naprave elektroenergetskega postroja, ki predstavljajo izvor ali ponor signalov (informacij) in omogoča vpogled v stanje nadziranega procesa. Pojem naprav v kontekstu modeliranja je abstrakcija konkretne tehnične naprave in se ne ozira na funkcijo in aplikacijo te naprave v konkretnem EE postroju ampak na informacijske značilnosti, ki opisujejo delovanje postroja. Primer tipov naprav EE postroja prikazuje tabela 4.

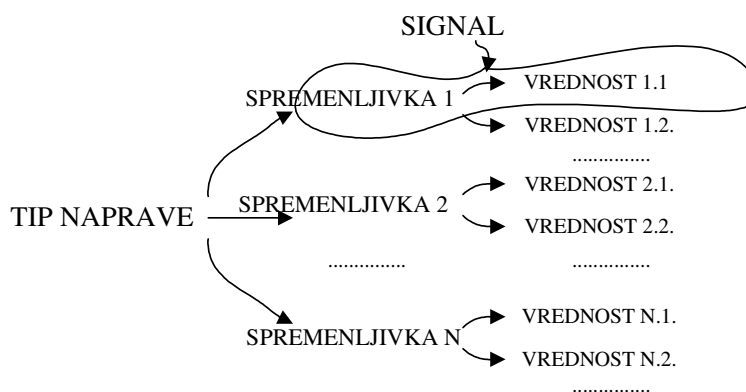
⁵⁴ device

Tabela 4: Primer tipov naprav v EE postroju

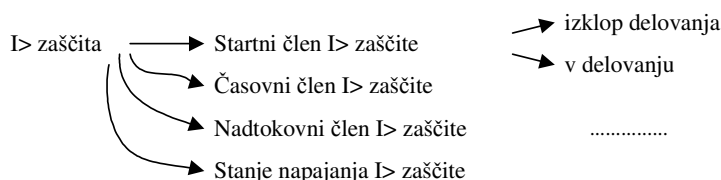
Koda	Tipi naprav
1.	izvodni ločilnik
2.	izvodni SN ločilnik
	
7.	odklopnik SF ₆
8.	malooljni odklopnik
	
14.	diferenčni rele ΔI
15.	nadtokovni zaščitni I> rele
	

V naslednjem koraku definiramo spremenljivke stanja, ki opisujejo delovanje posameznih tipov naprav. Slika 29 prikazuje modeliranje tipa naprave in njej pripadajočih spremenljivk. Spremenljivko lahko opredelimo kot kvantiteto, ki lahko zavzame katerokoli podano vrednost in s katero lahko opišemo delovanje naprave. V določenem časovnem oknu lahko spremenljivki pripišemo neko vrednost, ki odraža stanje naprave. Spremenljivka mora imeti vsaj dve vrednosti, da lahko spreminja svoje stanje. Kombinacija spremenljivke in njene vrednosti tvori signal, ki ga sproži prehod v določeno stanje naprave ali na podlagi katerega se izvrši sprememba stanja naprave, t.j. vrednost spremenljivke stanja.

Slika 29: Modeliranje tipa naprav



Primer:



Skupina signalov neke naprave tvori statični nabor vseh možnih signalov, ki opisujejo delovanje naprave. Primer tipov naprav in povezanih signalov za dve tehnični napravi EE postroja je prikazan v tabeli 5, kjer prvi stolpec vsebuje kodo za vsak tip naprave, drugi

identifikacijo tipa naprave, tretji pa signale sestavljene iz imena spremenljivk in njihovih vrednosti.

Tabela 5: Nabor signalov za tipe naprave

Nabor signalov za tipe naprav			
koda	Tip naprave	Signali	
		Ime spremenljivke	Vrednost
1	Izvodni ločilnik	Izvodni ločilnik - komanda	vklop izklop
		Izvodni ločilnik	vklopljen izklopljen vmesni položaj napaka
.....			
15	I> zaščita	Startni člen I> zaščite	v delovanju izklop delovanja
		Stanje napajanja I> zaščite	v redu napaka
		Časovni člen I> zaščite	v delovanju izklop delovanja
		Nadtokovni člen I> zaščite	v delovanju izklop delovanja
.....			

Statični nabor signalov za posamezne tipe naprav je izdelan neodvisno od posameznega konkretnega EE postroja in ga je zato mogoče aplicirati na različne tipe postrojev npr. postroje hidroelektrarn, termoelektrarn ali prenosnih ter distribucijskih omrežij.

Strukturo signalov lahko prikažemo v trodimenzionalni obliki kot vektor (primerjaj s sliko 8 podpoglavja 3.4.2.1), kjer so tipi naprav nanešeni na x os, spremenljivke stanja na y os, vrednosti spremenljivk stanja pa na z os. Iz slike 30 razberemo sledeče organizirani nabor signalov:

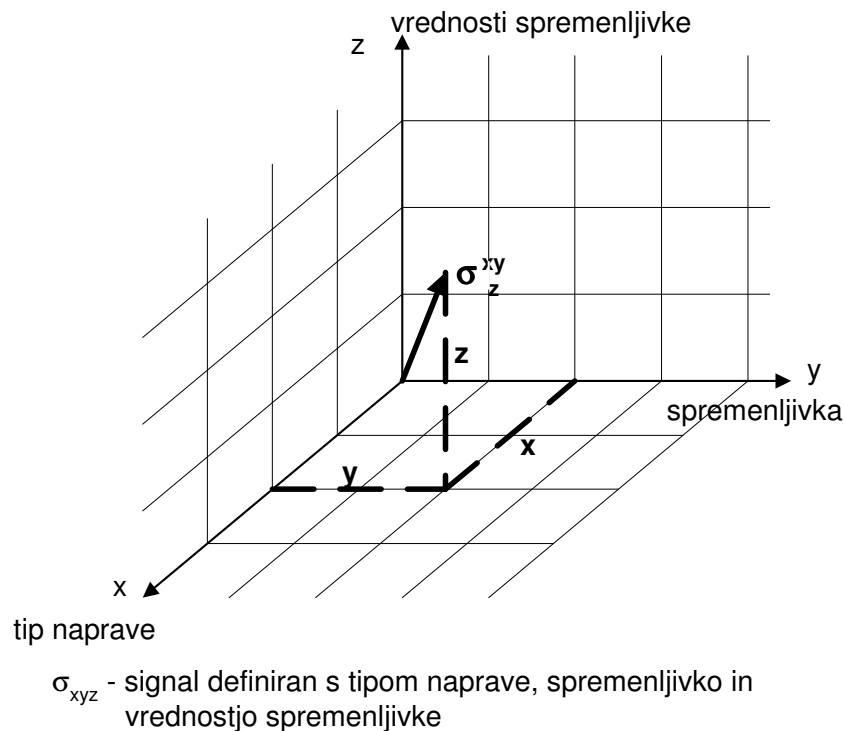
$$S_D = \{\sigma_{xyz}\}, \quad \text{kjer je } x=1,2,3,\dots,m$$

$$y=1,2,3,\dots,n$$

$$z=1,2,3,\dots,r \quad (2)$$

σ_{xyz} predstavlja signal definiran s tipom naprave, spremenljivko stanja in vrednostjo spremenljivke, števniki m pomeni število tipov naprav, n število spremenljivk stanja in r število vrednosti spremenljivk za vsako spremenljivko.

Slika 30: Aparaturna predstavitev signala



Vir: Šimunić, 2001, str. 36.

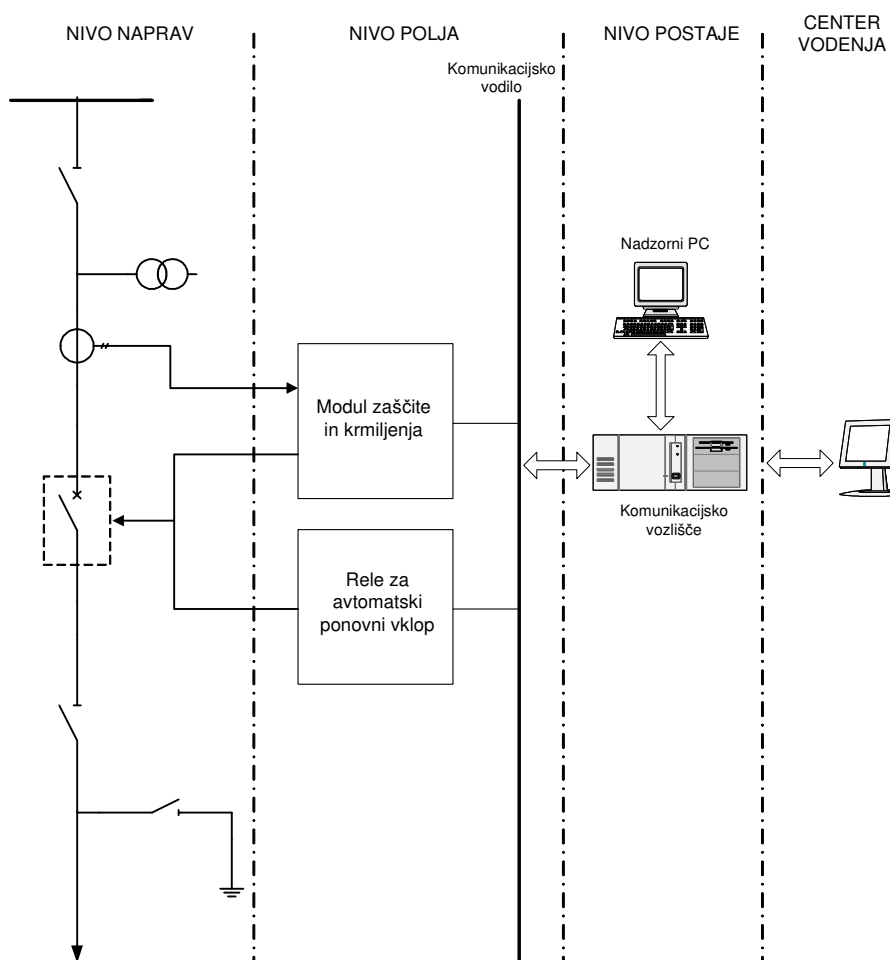
Če primerjamo enačbo (1) v podpoglavju 3.4.2.1, ki implicira funkcijsko modeliranje procesnih podatkov, z enačbo (2), ki implicira aparaturno modeliranje podatkov, lahko potegnemo sledeče zaključke:

- Formalno je strukturno-izrazni okvir za določen nabor procesnih signalov, t.j. S_{FT} in S_D , popolnoma analogen.
- V osnovi S_{FT} in S_D predstavljata popolnoma različna nabora procesnih podatkov in tako omogočata različen pristop k načrtovanju procesnih signalov.
- S_{FT} nabor je strogo vezan na tehnologijo posameznega EE postroja in ga ni mogoče direktno aplicirati na nekem drugem postroju.
- S_D nabor je generičen, t.j. splošen, tehnološko neodvisen, načrtovalsko odprt in tako uporaben za različne aplikacije.
- Procesni signali generirani preko aparaturnega modeliranja predstavljajo tehnološko neodvisno podatkovno bazo signalov, ki je enostavno razširljiva z novimi signali.

5.2.1.2 Tehnološka dekompozicija EE postroja

V predhodnem podpoglavju je kot prva zahteva v procesu modeliranja podatkov definiran model tipov naprav in njihovih stanj. Nadaljnja zahteva je definiranje modela celotnega EE postroja, ki naj omogoči nedvumljivo identifikacijo naprav glede na njihovo lokacijo znotraj njega samega. Identifikacija naprav glede na lokacijo mora torej slediti tehnološki strukturi EE postroja. Slika 31 prikazuje primer dela EE postroja in pripadajočih naprav ter povezave v nadrejeni center vodenja.

Slika 31: Primer dela EE postrojenja



Vir: Randić, 2000, str. 997.

Tudi manjši primerki EE postrojev so za zajem njihovega celotnega delovanja prekompleksni, zato je nujno opisati in modelirati interno funkcionalnost sistema tako, da se sistem razdeli na podsisteme. Dekompozicija sistema na podsisteme upošteva naslednje principe:

- Definirane morajo biti meje med podsistemi in načini njihove interakcije na teh mejah.
- Podsistemi morajo biti relativno elementarni in enostavni za opis notranjega delovanja.
- Podsistemi se ne smejo prekrivati.
- Celotni modelirani sistem mora biti zgrajen s kombiniranjem njegovih podsistemov.

Pogoju neprekrivanja med podsistemi je zadoščeno z vpeljavo konceptov tehnološke enote, tehnološkega polja, tehnološkega modula⁵⁵ in tehnološkega podmodula (slika 32).

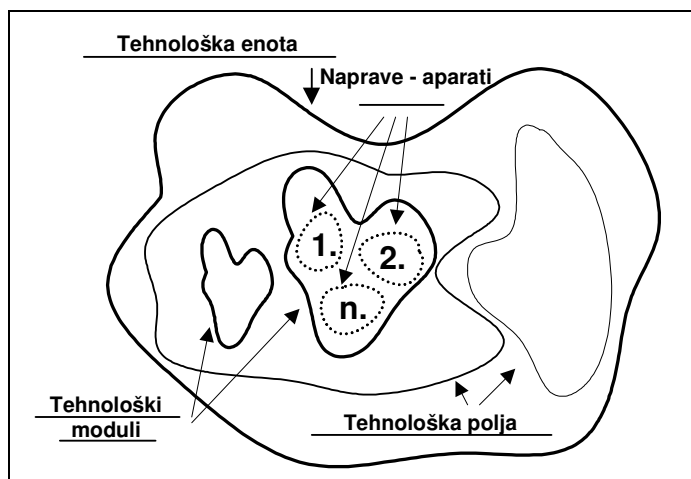
Tehnološka enota predstavlja skupino naprav v podsistemu EEP, ki izvajajo neko kompleksno funkcijo (komandni prostor, stikališče EEP itd.).

Tehnološko polje je del tehnološke enote, kjer se ločujejo npr. napetostni nivoji EEP (400 kV, 220 kV, 110 kV, 20 kV, itd.)

⁵⁵ Pojem je vsebinsko različen od uporabljenega pojma v podpoglavju 3.4.2.1, kjer tehnološki modul predstavlja izvor in ponor podatkov v kontekstu funkcijskega modeliranja podatkov.

Tehnološki modul predstavlja v okviru tehnološkega polja npr. posamezne napajalne izvode iz EEP (npr. kablovod 20 kV RTP Gorica – RTP Ajdovščina). Po potrebi so tehnološki moduli nadalje razdeljeni na tehnološke podmodule.

Slika 32: Dekompozicija sistema



Vir: Šimunić, 2001, str. 36.

Slika 32 prikazuje razdelitev enega dela EEP (tehnološke enote) na tehnološka polja, tehnološke module in naprave. Določitev tehnoloških polj in modulov je predmet modeliranja EEP. Če so tehnološki moduli izbrani preozko, bo takšnih modulov veliko število in obstaja možnost prekrivanja med njimi, kar ima za posledico, da ne bo zadoščeno konsistentnemu kvalitativnemu modeliranju. Če pa so tehnološki moduli preširoko izbrani, se lahko zgodi, da signali ne bodo natančno definirani in alocirani. To se dogodi predvsem takrat, ko v procesu znotraj enega tehnološkega modula nastopa večje število naprav istega tipa. V taki situaciji je potrebno oddvojiti manjše podmodule znotraj osnovnega tehnološkega modula in tako nedvoumno določiti nabor signalov.

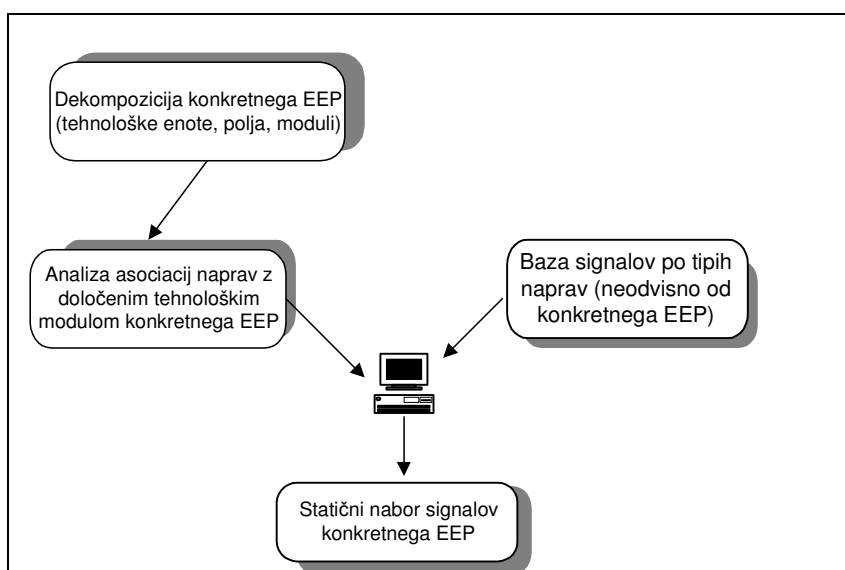
Na opisani osnovi je sedaj mogoče razpoznati procedure za generiranje nabora statičnih procesnih signalov za določen EEP. Procedura vključuje najprej identifikacijo tehnoloških enot, polj in modulov v EEP. Nato naprave asociiramo z moduli, ki jim pripadajo. Tabela 6 prikazuje princip asociacije naprav s posameznimi tehnološkimi moduli.

Tabela 6: Asociacija naprav s tehnološkimi moduli

koda	Tehnološki modul	Asocirane naprave (kode)
9	20 kV kablovod Gorica 1	1, 3, 7, 32, 33, 34, 36, 37, 38, 64, 74, 94, 97, 111

Vsaka konkretna naprava ima svoj abstraktni tip, zato signali, ki jih naprava pošilja ali sprejema, ustrezajo naboru signalov abstraktnega tipa. Procedura za generiranje statičnega nabora signalov je za EEP predstavljena na sliki 33, izpis tako generiranih signalov pa je prikazan v tabeli 7.

Slika 33: Procedura generiranja signalov



Vir: Šimunić, 2001, str. 37.

Tabela 7: Primer izpisa generiranih signalov

EPP GORICA	20kV KBV Gorica 1	Izvodni ločilnik - komanda	vklop
EPP GORICA	20kV KBV Gorica 1	Izvodni ločilnik - komanda	izklop
EPP GORICA	20kV KBV Gorica 1	Izvodni ločilnik	vklopljen
EPP GORICA	20kV KBV Gorica 1	Izvodni ločilnik	izklopljen
EPP GORICA	20kV KBV Gorica 1	Izvodni ločilnik	vmesni položaj
EPP GORICA	20kV KBV Gorica 1	Izvodni ločilnik	napaka
EPP GORICA	20kV KBV Grgar	Izvodni ločilnik - komanda	vklop
EPP GORICA	20kV KBV Grgar	Izvodni ločilnik - komanda	izklop
EPP GORICA	20kV KBV Grgar	Izvodni ločilnik	vklopljen
EPP GORICA	20kV KBV Grgar	Izvodni ločilnik	izklopljen
EPP GORICA	20kV KBV Grgar	Izvodni ločilnik	vmesni položaj
EPP GORICA	20kV KBV Grgar	Izvodni ločilnik	napaka
EPP GORICA	20kV KBV Gorica 1	I> zaščita startni člen	OK
EPP GORICA	20kV KBV Gorica 1	I> zaščita startni člen	okvara
EPP GORICA	20kV KBV Gorica 1	I> zaščita stanje napajanja	OK
EPP GORICA	20kV KBV Gorica 1	I> zaščita stanje napajanja	okvara
EPP GORICA	20kV KBV Gorica 1	I> zaščita časovni člen	OK
EPP GORICA	20kV KBV Gorica 1	I> zaščita časovni člen	okvara
EPP GORICA	20kV KBV Gorica 1	I> zaščita nadtokovni člen	OK
EPP GORICA	20kV KBV Gorica 1	I> zaščita nadtokovni člen	okvara

5.2.1.3 Definiranje funkcij in lokacij uporabe signalov

Potem ko je izvedeno generiranje statičnega nabora signalov, je naslednji korak v postopku modeliranja določitev lokacij uporabe signalov. Če kot primer EEP uporabimo konkretno razdelilno transformatorsko postajo (RTP), potem postroj razdelimo na sledeče tehnološke enote: stikališče, komandni prostor, prostor pomožnih naprav, itd. Lokacije uporabe signalov za stikališče so tako krmilna omarica (KO), krmilno-zaščitni panel (KZP), krmilno-relejna omara (KRO), itd; za komandni prostor sinoptična plošča (SO), nadzorni računalnik (NPC), itd; za prostor pomožnih naprav požarni panel (PP), panel napajanja lastne rabe (PL), itd. Dodatno je potrebno kot lokacije uporabe signalov definirati še oddaljene nadzorne sisteme, v našem primeru so to distribucijski center vodenja DEES in njegova oddaljena dispečerska mesta. S tem korakom je zaključen postopek aparaturno-orientiranega modeliranja EEP. Tabela 8 prikazuje signale nekega EEP in lokacije njihove uporabe.

Tabela 8: Signali in lokacije njihove uporabe

	Signali EEP	KO	KZP	KRO	SO	NPC	PP	PL	DCV
	RTP_KBV 20kV Gorica1_ izvodni ločilnik - stanje _ vklopljen		+	+		+			+
	RTP_KBV 20kV Gorica1_ izvodni ločilnik - stanje _ izklopljen		+	+		+			+
	RTP_KBV 20kV Gorica1_ izvodni ločilnik - stanje _ vmesni		+	+		+			+
	RTP_KBV 20kV Gorica1_ izvodni ločilnik - stanje _ napaka		+	+		+			+
									
	RTP_DV 110kV Solkan_napajanje_ krmiljenje Q0_izpad	+		+	+	+		+	+
	RTP_DV 110kV Solkan_napajanje_ krmiljenje Q0_ok	+		+	+	+		+	+
									
	RTP_komandni prostor_požarna naprava_ modul_alarm					+	+		+
	RTP_komandni prostor_požarna naprava_ modul_ok					+	+		+
									

Kot zaključek povzamemo, da se opisani aparaturno-orientirani postopek modeliranja EEP izvaja v treh fazah:

- Rezultat prve faze je generičen nabor procesnih signalov po tipu naprav.
- Rezultat druge faze je konkreten nabor procesnih signalov na nivoju EEP.
- Rezultat tretje faze so definirane lokacije za vsak posamezen signal.

5.2.2 Preslikava aparaturnega modeliranja v objektno paradigmo

Bistvo aparaturnega informacijskega modela je asociacija naprav EEP s tehnološkimi moduli EEP. Naprave pošiljajo in sprejemajo različne signale. Aparaturno modeliranje temelji na abstrakciji naprav in ga je mogoče aplicirati na informacijski model poljubnega EEP, v nasprotju s funkcijskim, ki takšne fleksibilnosti ne podpira.

V nadaljevanju se bom osredotočil na prikaz pomembne značilnosti aparaturnega modeliranja, in sicer te, da modeliranje, ki temelji na napravi kot temeljnemu viru in ponoru informacij, v svojem bistvu odsluži objektno-orientirani pogled na EEP in na njegove procese. Aparaturni model bo tako nadomeščen z objektno-orientiranim modelom podatkov, ki se izmenjujejo v procesu EEP.

5.2.2.1 Analogija med aparturnim in objektnim modelom EEP

Objektni model predstavitve podatkov temelji na konceptih objektov, razredov, atributov, sporočil, stanj, operacij, obnašanja, itd. Objekt je konkreten pojav neke abstrakcije in je tako temeljna entiteta v modelu z natančno določenimi mejami in identiteto ter vključuje njegovo stanje in obnašanje. Ker v modelu EEP naprave predstavljajo vire in ponore informacij, lahko posledično ugotovimo, da bodo v objektnem informacijskem modelu EEP najpomembnejši objekti praviloma abstrakcije realnih naprav EEP. Ti objekti bodo predstavljali abstrakcije tehničnih naprav, ki so dejansko vgrajene v nekem EEP.

Identiteto objektov lahko določimo na podlagi njihove lokacije tako, da sledimo tehnološki dekompoziciji opisani v prejšnjem podpoglavju (slika 32) npr. tako, da ima identifikator objekta, ki predstavlja abstrakcijo izvodnega ločilnika (IL) priključenega na 110 kV zbiralke št. 1 daljnovodnega polja (DVP) Solkan razdelilne transformatorske postaje (RTP) Gorica, sledečo strukturo:

RTPGorica.110kV.DVPSolkan.IL1.

Splošno zapisano ima objekt, ki predstavlja napravo EEP, identifikator s strukturo:

EEP.NapetostniNivoID.TehnološkiModulID.NapravaID.

V določenem trenutku časa se vsak objekt nahaja v nekem stanju. V celotni življenjski dobi obstoja objekt zadošča določenim stanjem (npr. izvodni ločilnik je vklopljen), izvaja določeno aktivnost (npr. izvodni ločilnik se vklaplja) ali pa čaka na določen dogodek (npr. izvodni ločilnik je pravkar sprejel signal oziroma komando *Izklop*). Stanje objekta je torej definirano s trenutno kombinacijo vrednosti njegovih spremenljivk stanja, t.j. atributov. Stanje objekta je tudi odvisno od stanja drugih objektov, če obstaja povezava med njimi. V procesu EEP vedno obstaja povezava med izvodnim ločilnikom⁵⁶ in odklopnikom⁵⁷. Poleg svojih osnovnih stanj (*vklopljen*, *izklopljen*, *vmesni položaj*, *napaka*) je namreč lahko izvodni ločilnik v stanju *blokiran*, glede na stanje z njim povezanega odklopnika. Enostavna abstrakcija naprave izvodnega ločilnika lahko rezultira v tip objekta (razred) *Izvodni ločilnik* z atributom *položaj*, ki lahko zavzame diskretne vrednosti (*vklopljen*, *izklopljen*, *vmesni položaj*).

Objekti so poleg tega, da se nahajajo v določenem stanju, tudi odgovorni za obnašanje. Obnašanje objekta lahko opredelimo kot karkoli, kar je zaznano ali pa je rezultat nekega dogodka (signal, pretek časa, sprememba stanja, zahteva za izvedbo določene operacije, ipd.). Operacija torej predstavlja storitev, ki je vkomponirana v objektu samem. Kot primer navajam zahtevo po vklopu objekta tipa *Izvodni ločilnik* tako, da se objektu pošlje zahteva za izvedbo operacije *Vklop()*.

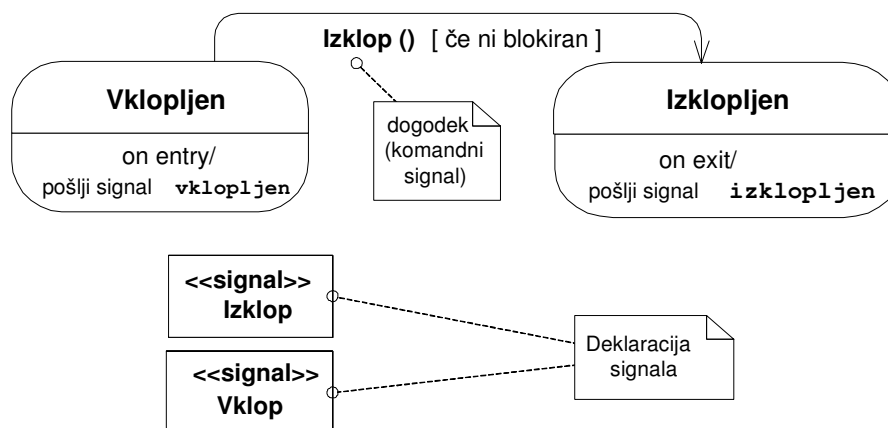
V objektno-orientiranem modelu EEP so najpogostejši dogodki spremembe stanj objektov in pojav določenih signalov. Pojav oziroma odposlanje signala je potemtakem neke vrste dogodek. Signali se izmenjujejo med objekti, kjer eni nastopajo kot pošiljatelji (npr. objekti kot naprave EEP), drugi pa nastopajo kot sprejemniki signalov (npr. objekti, ki predstavljajo lokacije uporabe teh signalov). Objekt tipa ali razreda *Izvodni ločilnik* lahko sprejme signal *Izklop()* od nadzornega objekta. Nadzorni objekt predstavlja abstrakcijo enega dela

⁵⁶ Izvodni ločilnik služi samo vidnemu ločevanju energetskih tokokrogov, ne odklaplja pod obremenitvijo.

⁵⁷ Odklopnik služi odklapljanju moči pod obremenitvijo in v primeru odklopa kratkega stika v omrežju.

nadzornega sistema EEP. Signal bo povzročil spremembo stanja izvodnega ločilnika, če je le-ta v stanju oziroma položaju *vklopljen*. Potem ko je spremenil stanje iz *vklopljen* v *izklopljen*, bo izvodni ločilnik odposlal informacijski signal o spremembi stanja nadzornemu objektu. Opisani primer obnašanja prikazuje diagram stanj na sliki 34, kjer je razvidno, da so v objektnem modelu lahko tudi signali obravnavani kot objekti, in sicer kot primerki določenega tipa signalov, ki se medsebojno izmenjujejo med parom objektov pošiljatelj - sprejemnik.

Slika 34: Diagram prehoda stanj za objekt *Izvodni ločilnik*



Vir: Šimunić, 2001, str. 38.

Koncept objektnega tipa (razreda) predstavlja nabor objektov iste vrste, ki delijo enake atribute, operacije in povezave z drugimi tipi objektov.

Tabela 9 ilustrira analogijo oziroma možno preslikavo med koncepti aparaturnega in objektno-orientiranega modeliranja EEP.

Tabela 9: Preslikava aparaturnega v objektni koncept modeliranja

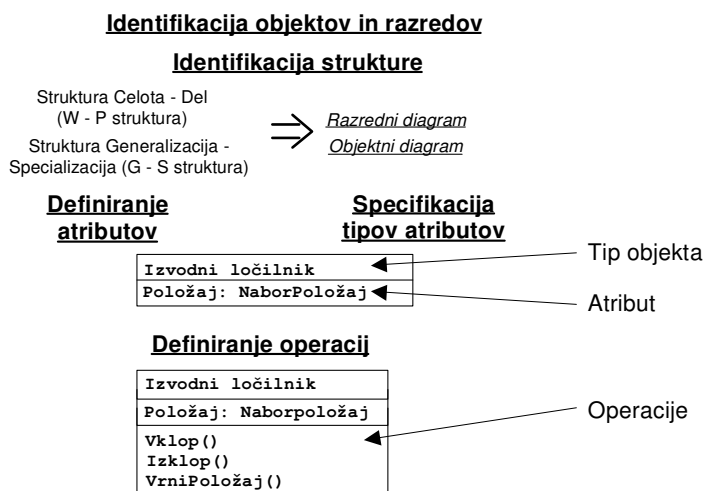
Koncepti aparaturno-orientiranega modeliranja	Koncepti objektno-orientiranega modeliranja
naprava	objekt
tip naprave	tip objekta (razred)
spremenljivka	atribut
vrednost spremenljivke	vrednost atributa
signal	sporočilo
stanje naprave	stanje objekta
komanda	operacija
tehnološko polje, modul	agregacijska struktura agregat-deli
lokacija uporabe signala	sprejemnik sporočila (objekt)

Vir: Šimunić, 1998, str. 442.

5.2.2.2 Objektno-orientirana strategija modeliranja

Osnovne korake objektno-orientiranega modeliranja EEP prikazuje slika 35. Prvi korak *Identifikacija objektov in njihovih tipov (razredov)* vključuje identifikacijo objektov kot abstrakcij konkretnih tehničnih naprav, ki ohranjajo svoje značilnosti kot take ne glede na njihovo funkcijo v EEP, obenem pa upoštevajoč informacijske in procesne značilnosti, s katerimi je njihovo procesno delovanje v EEP opisano.

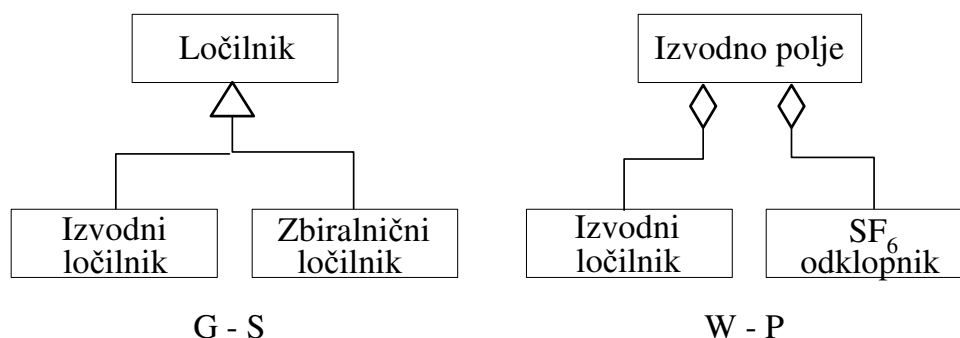
Slika 35: Koraki objektne strategije modeliranja EEP



Vir: Šimunić, 1998, str. 442.

Nadaljnji korak (slika 36) je *Identifikacija strukture*, ki vključuje identifikacijo strukture tipa generalizacija – specializacija (G – S) in tipa celota – del (W – P).

Slika 36: Primer strukture G – S in W – P



Vir: Šimunić, 1998, str. 442.

Definiranje atributov je naslednji korak, kjer identificiramo in poimenujemo bistvene značilnosti naprav. Vrednosti atributov so odvisne od njihovih tipov, ti pa so znotraj modela specificirani z neko standardno notacijo npr. UML. *Definiranje operacij* je korak, ko objektom pripišemo storitve, ki so na razpolago podsistemom ali objektom zunaj njih samih. Operacije preko implementiranih metod predstavljajo stik oziroma interakcijo objekta z zunanjimi

podsystemi. Zahteve po storitvah, ki jih zahtevajo zunanji podsystemi oziroma objekti, so objektom, ki te storitve nudijo, posredovani preko sistema izmenjave sporočil oziroma signalov. Kot primer lahko objekt tipa *Izvodni ločilnik (IL1)* prejme sporočilo (signal):

RTPGorica.110kV.DVPSolkan.IL1.Vklop

Sporočilo (signal) o spremembi stanja objekta, ki ga objekt pošlje kontrolnim zunanjim objektom, je specificirano v kontekstu diagrama stanj, ki je izdelan za vsak objekt posebej. Na sliki 34 je prikazan le del diagrama stanj za razred *Izvodni ločilnik*. Celoten diagram stanj za ta razred bi moral specificirati vsaj tri signale, ki se generirajo ob spremembi stanja objektov razreda:

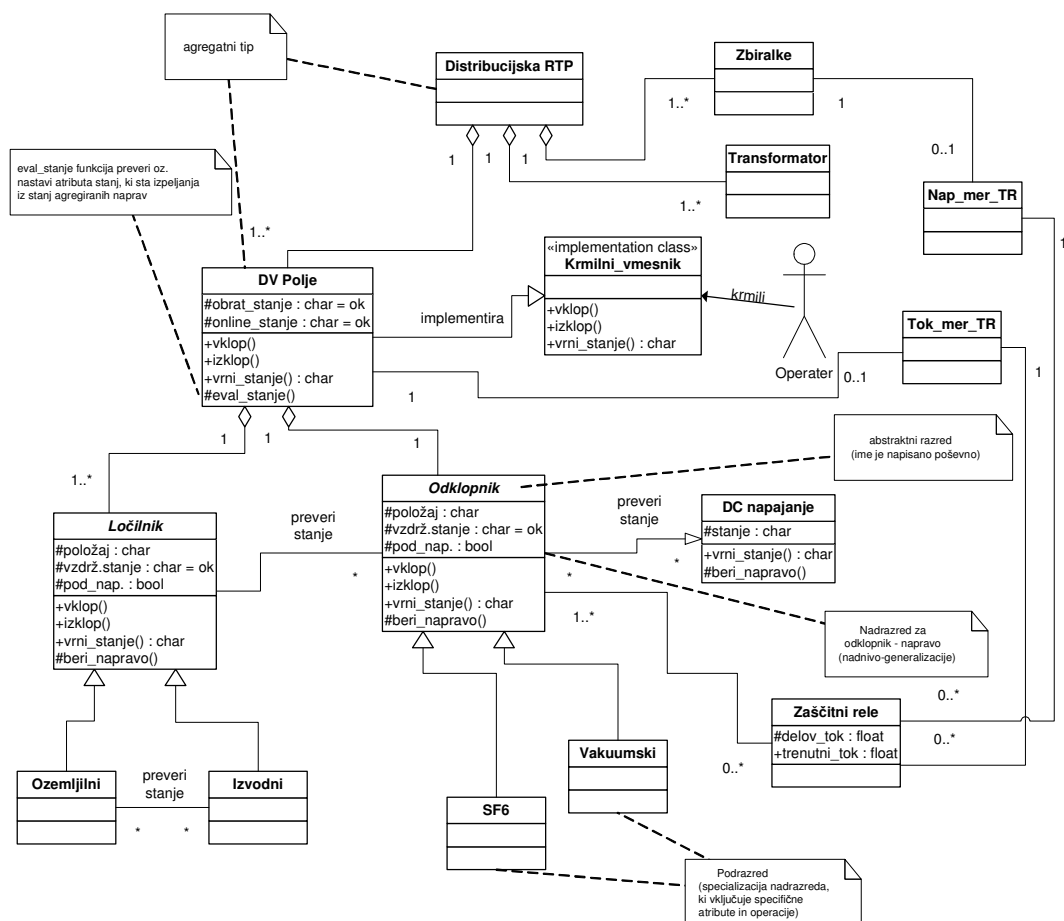
RTPGorica.110kV.DVPSolkan.IL1.Vklopjen

RTPGorica.110kV.DVPSolkan.IL1.Izklopljen

RTPGorica.110kV.DVPSolkan.IL1.VmesniPoložaj

Do tukaj nakazani primeri ilustrirajo samo bistvene aspekte za vzpostavitev relacije med splošnim objektnim pristopom k modeliranju in aparaturnim modeliranjem procesnih podatkov v konkretnem EEP. Primer celostne objektno obravnave strukture nekega EEP (Razdelilne transformatorske postaje) prikazuje slika 37.

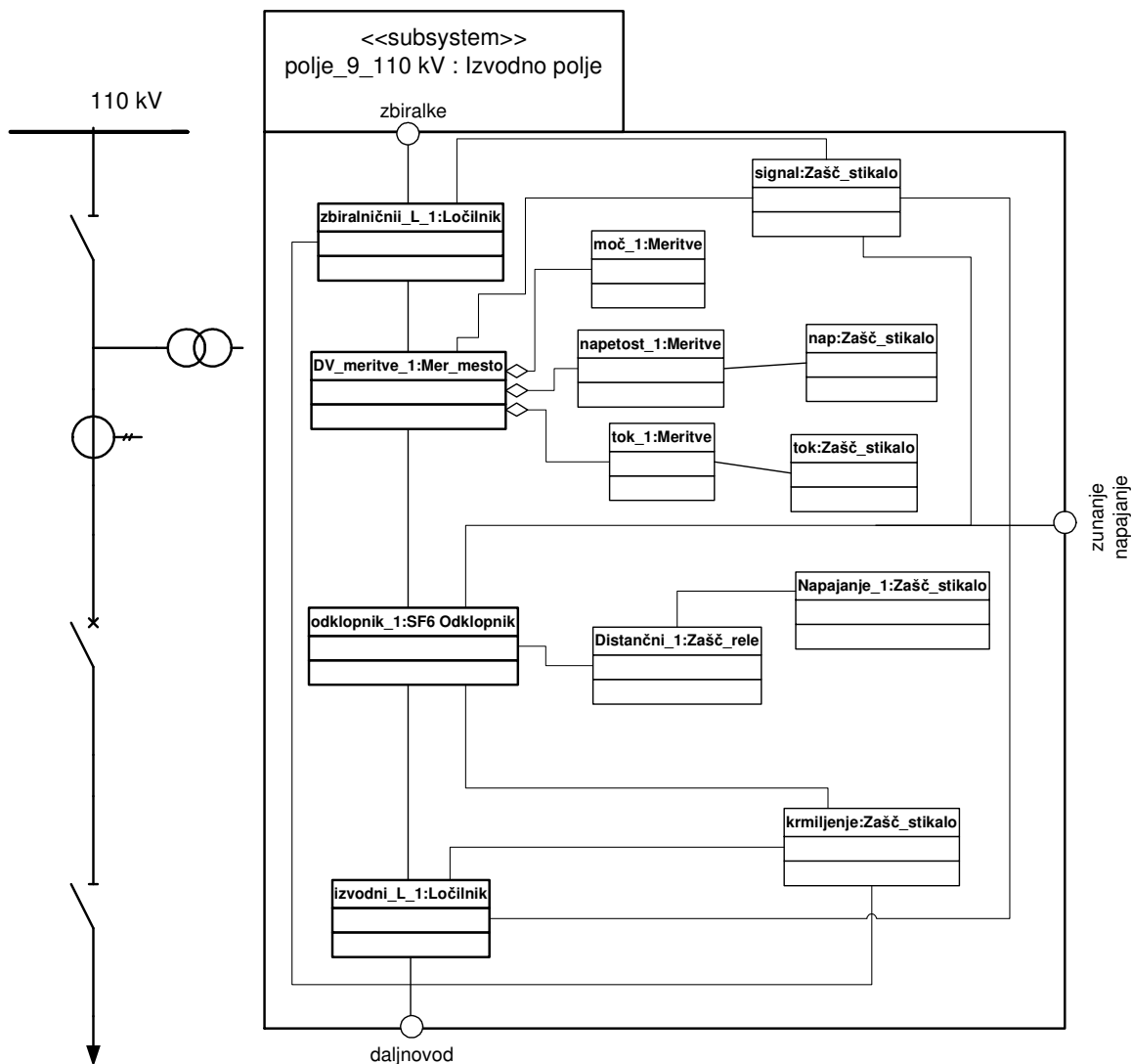
Slika 37: Primer razrednega diagrama EEP



Vir: Gudac, 1999, str. 172.

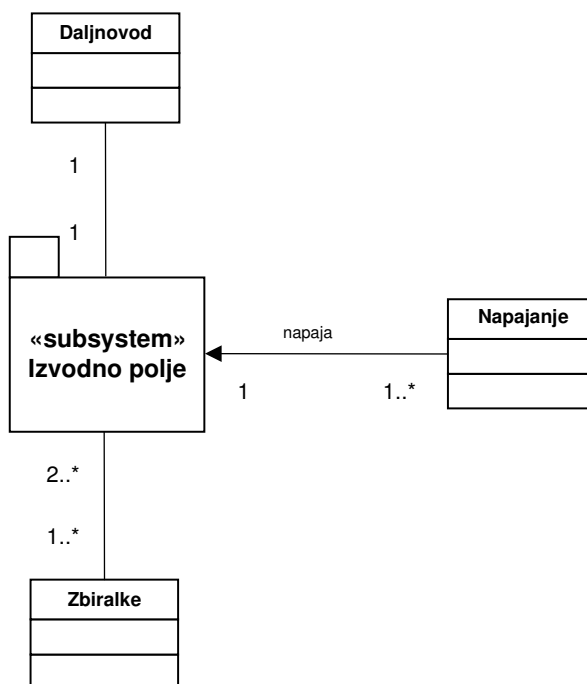
Podrobno razdelani razredni diagram na sliki 37 lahko vključimo in prikažemo tudi v kontekstu razdelitve celotnega sistema v podsisteme. V nadaljevanju bo na sliki 38 prikazan podsistem *Izvodno polje*, kjer bo razvidna povezava konkretnih elementov (naprav) nekega visokonapetostnega izvodnega polja z razredi objektov, ki v prikazanem modelu predstavljajo njihovo abstrakcijo. Tak prikaz osredotoči pozornost na bistvene sestavne dele obravnavanega modela. Odločimo se, da v podsistem *Izvodno polje* vključimo vse elemente, ki kakorkoli neposredno interagirajo v kontekstu izvodnega polja (slika 39). Odločitev, kje postaviti meje nekega podsistema, je vedno v rokah analitika in načrtovalca modela.

Slika 38: Podsistem Izvodno polje



Vir: Đurović, 1999, str. 5.

Slika 39: Prikaz povezav podsistema EEP



Vir: Đurović, 1999, str. 3.

5.3 Objektno modeliranje delovnih procesov za podporo poslovanju

Podjetja za distribucijo električne energije oskrbujejo odjemalce na nizki (0.4 kV) in srednji (20 kV) napetosti. Oblikovana so kot delniške družbe z zaenkrat večinsko kapitalsko udeležbo države, hkrati pa imajo tudi status javnih podjetij. Kot taka so trenutno še precej toga glede sprejemanja novih zahtev trga. Električna energija namreč z deregulacijo dejavnosti postaja tržno blago, kar sili posamezna podjetja, da investirajo v nove, ljudem in okolju prijazne tehnologije. Z možnostmi, ki jih danes z novimi pristopi nudijo programska, strojna in TK oprema, je mogoče funkcije zgolj tehničnega vodenja nadgraditi s funkcijami, ki nudijo podporo poslovnim odločitvam in omogočajo poleg zagotovitve kvalitetnejše dobave električne energije tudi prijaznejši odnos do odjemalcev. Nedobavljena električna energija je danes še vedno težko ovrednotena, težko pa je tudi predvideti, kako bi posodobitve vplivale na zmanjševanje stroškov poslovanja. Edina dejavnika, ki ju je mogoče finančno ovrednotiti, sta energetske izgube v omrežju ter izpad prihodka zaradi daljše ali krajše prekinitve dobave električne energije. Še vedno velja, da četudi uspemo z investicijskim programom dokazati ekonomsko upravičenost investicije, je njen pozitiven učinek še vedno preveč odvisen od tehničnega obvladovanja sistema. Natančnejšo ekonomsko upravičenost bi bilo mogoče dokazati takrat, ko bodo dokončno znani pogoji gospodarjenja v dereguliranem okolju (organizacija, lastniška struktura, tarife, cena uporabe omrežja, penalni faktorji itd.).

V nadaljevanju bo prikazana konkretna aplikacija sistema za upravljanje z okvarami v DEES kot enega izmed podpornih orodij v centrih vodenja DEES z uporabo objektnih konceptov za modeliranje sistemov in procesov ter UML notacije.

5.3.1 Problemski dokument

Po definiciji opravljajo distribucijski centri vodenja naloge vodenja DEES, ki obsegajo nadzor, odločanje in krmiljenje naprav. Te osnovne operativno-procesne funkcije se izvajajo preko SCADA programske opreme. Razvoj in implementacija novih informacijskih tehnologij omogoča dopolnitev in nadgraditev osnovne funkcionalnosti vodenja, tako da je mogoče avtomatizirati in informatizirati tudi operativno-poslovne funkcije, ki so jih operaterji v DCV do tega trenutka opravljali »peš«, to je ročno, brez uporabe računalnikov. Med te operativno-poslovne funkcije sodijo sprejemanje klicev odjemalcev in nudenje informacij o stanju v DEES, evidentiranje (beleženje) okvar pri odjemalcih, sprejemanje zahtev po odklopih naprav zaradi planiranih del, potrjevanje in odobravanje odklopov naprav, priprava navodil za delo (delovni programi, delovni nalogi), upravljanje terenskih vzdrževalnih ekip, sledenje poteka planiranih in havarijskih del, priprava internih poročil o okvarah in stanju DEES, priprava poročil za zunanje inšpekcijske in regulatorne službe itd. Nekatere od teh bodo v magistrskem delu tudi obdelane. Na področju lociranja in odprave okvar v DEES bi bil načrtovani sistem lahko zameetek ekspertnega sistema za pomoč operaterjem v DCV. Poleg tega v prihodnosti predvidevam nadgradnjo takega sistema s sistemom za (avtomatsko) obdelavo klicev odjemalcev, kar bi dodatno razbremenilo operaterja v DCV, informatizacija izmenjave podatkov o okvarah med obema sistemoma pa bi hkrati pomenila dvig kvalitete storitev za odjemalce električne energije.

Osnovna funkcionalnost načrtovanega podsistema bo omogočila natančno vodenje evidence okvar DEES, pregled okvar glede na izbrane kriterije, pripravo poročil o okvarah v povezavi z izbranimi parametri itd. Vsak dogodek v zvezi z evidentirano okvaro bo v sistemu pustil sled, tako da bo možno v vsakem trenutku spremljati življenjski cikel posamezne okvare in življenjski cikel postopkov, ki so povezani z odpravo okvare.

Za razumevanje funkcionalnosti načrtovanega sistema podajam v nadaljevanju opis možnih scenarijev v zvezi z življenjskim ciklom okvare in postopkov za njeno odpravo:

- Odjemalec pokliče na dežurno številko za prijavo okvare v EES. Operater v DCV zabeleži podatke o odjemalcu in vrsti okvare.
- V primeru že evidentirane okvare na področju kličočega, bodisi iz predhodnih klicev odjemalcev bodisi iz SCADA sistema, operater o tem in o postopkih v zvezi z njeno odpravo obvesti odjemalca, sicer se prijava okvare evidentira kot nov dogodek v DEES in kot taka čaka na nadaljnjo obdelavo oziroma korelacijo. V vsakem primeru se odjemalčev klic zabeleži v sistem, lahko se izda tudi servisna zahteva.
- V primeru nadaljnjih klicev odjemalcev sistem izvede korelacijo med posameznimi klici. Če klici korelirajo, potem evidentira področje, od koder prihajajo klici, kot področje okvare DEES. Operater na podlagi izvedene korelacije in točne lokalizacije okvare sproži pripravo servisnega naloga za odpravo okvare v smislu izdaje navodila za delo s podrobnimi stikalnimi navodili.
- V primeru, da v določenem časovnem oknu ni novih klicev s področja prvega kličočega, sistem evidentira okvaro pri odjemalcu, ki je klical. Operater na podlagi odjemalčeve prijave (servisne zahteve) sproži pripravo servisnega naloga v smislu izdaje navodila za odpravo okvare pri odjemalcu.

- Operater dodeli servisni nalog za odpravo okvare terenski posadki (ekipi) oziroma dežurnemu vzdrževalcu. Ekipa oziroma vzdrževalec opravi zahtevano delo in javi operaterju v DCV, da je okvara odpravljena. Vzpostavljeno je normalno obratovalno stanje.
- Operater v DCV zabeleži, da je vzpostavljeno prvotno stanje, zaključi servisni nalog. Hkrati se zaključijo tudi vse servisne zahteve, ki so vezane na predmetni servisni nalog. Okvara, ki je bila vzrok za servisne zahteve odjemlancev in izdajo servisnih nalogov, preide ob zaključku pripadajočih servisnih nalogov v stanje rešeno.

V primeru okvare DEES, ki jo avtomatsko zazna SCADA sistem vodenja, se okvara avtomatsko evidentira kot nov dogodek. Operater sproži postopek lokalizacije okvare in pripravo servisnega naloga za njeno odpravo. Nadaljnji postopek je enak kot že opisano.

5.3.2 Objektno UML modeliranje procesov

Poleg možnosti, ki so bile opisane v prejšnjem poglavju, UML omogoča vizualno predstavitev poslovnih in drugih sistemov v obliki diagramov primerov uporabe (funkcionalni model), diagramov statične strukture sistema (razredni in objektni model) in diagramov obnašanja sistema (dinamični model). Omogoča analizo in načrtovanje procesov ter je neodvisen od uporabe programskih jezikov za implementacijo.

Diagrami primerov uporabe prikazujejo komunikacijo med akterji (uporabniki, zunanji sistemi) in obravnavanim sistemom. Gradniki diagramov primerov uporabe so primeri uporabe, povezave med njimi in akterji. Diagrami primerov uporabe definirajo meje obravnavanih sistemov in njihovo poglavitno funkcionalnost ter so orodje za zbiranje zahtev.

Diagrami statične strukture sistema predstavljajo diagrame razredov in objektov ter relacij med njimi. Diagrami objektov so implementacije diagramov razredov in prikazujejo primerke, ki so skladni z diagrami razredov.

Diagrami obnašanja sistema prikazujejo dinamične informacije in dogodke, ki opisujejo časovno obnašanje. Obnašanje objektov prikazujejo diagrami stanj, diagrami aktivnosti, diagrami zaporedja in diagrami sodelovanja.

Analizo procesov in načrtovanje informacijskega sistema opravljamo iterativno, in sicer tako, da praviloma od splošne analize prehajamo k podrobni, hkrati pa rezultate analize prevajamo v model analiziranega procesa.

5.3.2.1 Diagrami primerov uporabe

Uporaba diagramov primerov uporabe omogoča, da zajamemo le tisti del sistema, ki nas trenutno zanima, saj opišemo njegovo partikularno delovanje in okolje. Na celoten proces gledamo kot na množico povezanih procesov, ki jih modeliramo s pomočjo primerov uporabe. Hkrati modeliramo tudi okolje procesov, in sicer kot množico akterjev, ki sodelujejo v primerih uporabe. Akterji so v splošnem lahko posamezni uporabniki sistema, naprave ali

drugi sistemi, ki z opisovanim sistemom komunicirajo, izmenjujejo informacije in iz njega pridobivajo rezultate.

Za potrebe sintakse primerov uporabe najprej na osnovi problemskega dokumenta opredelimo akterje, ki so v interakciji s sistemom. V osnovi so to naslednji:

- *Operater DCV* je zaposleni v DCV, upravljalec omrežja
- *Vodja DCV* je vodja operaterjev, odgovorni za delo
- *Poslovodstvo* je zaposleni, ki uporablja storitve DCV za notranje potrebe
- *Vzdrževalec* je terenski delavec, zadolžen za vzdrževanje in odpravo okvar na omrežju
- *Energetik* je zaposleni v okviru DCV, zadolžen za energetske analize
- *Tarifni odjemalec* je fizična ali pravna oseba, priključena na elektroenergetsko omrežje, ki nima možnosti izbire dobavitelja in je podvržena tarifnemu sistemu, ki ga določi vlada RS
- *Upravičeni odjemalec* je fizična ali pravna oseba, priključena na elektroenergetsko omrežje, ki ima možnost proste izbire dobavitelja na trgu
- *SCADA* je zunanji nadzorni sistem za vodenje omrežja
- *CIS* je zunanji informacijski sistem s podatki o odjemalcih
- *CALLC* je zunanji klicni center – sistem za upravljanje s klici odjemalcev
- *EXPERT* je zunanji ekspertni sistem za podporo odpravljanju okvar na omrežju
- *GIS* je zunanji geografski informacijski sistem

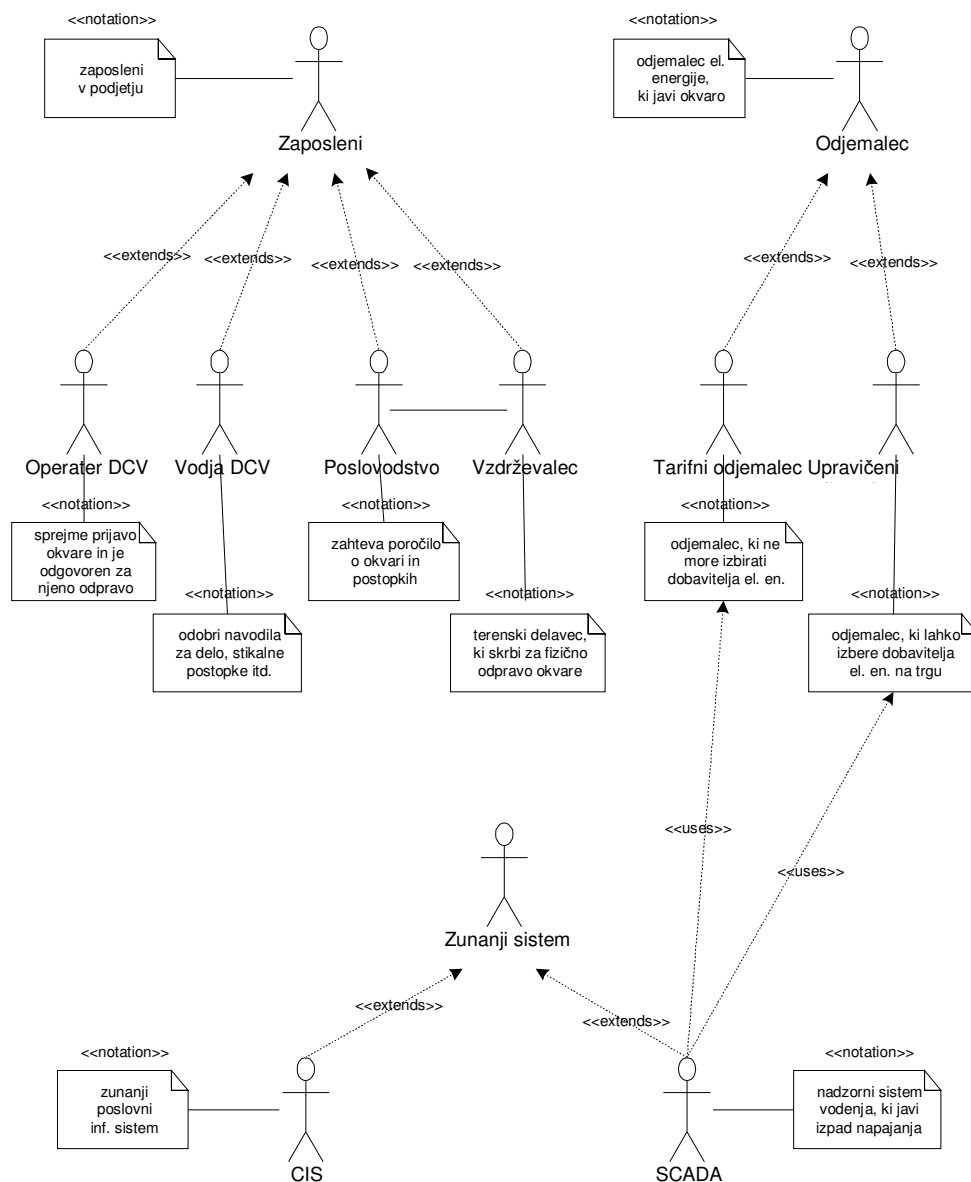
Prvi štirje navedeni predstavljajo kadrovske vloge, ki jih imajo zaposleni v podjetju za distribucijo električne energije. Akter Odjemalec je zunanji poslovni partner, ostali so zunanji sistemi, s katerimi je obravnavani sistem v interakciji. Za potrebe predmetnega dela se bom omejil le na nekatere od njih (slika 40). Funkcionalnosti, pri katerih sodelujejo ostali akterji (CALLC, EXPERT, GIS), presegajo zahteve tega teksta, zato bodo v nadaljevanju samo omenjene kot možne nadgradnje obravnavanega sistema.

Glede na zapisano v problemskem dokumentu lahko upravljanje z okvarami v DEES obravnavamo po scenariju, ko okvaro prijavi odjemalec ali več odjemalcev. Po tem scenariju predvidimo naslednje štiri temeljne primere uporabe (slika 41) za funkcionalnost obravnavanega sistema:

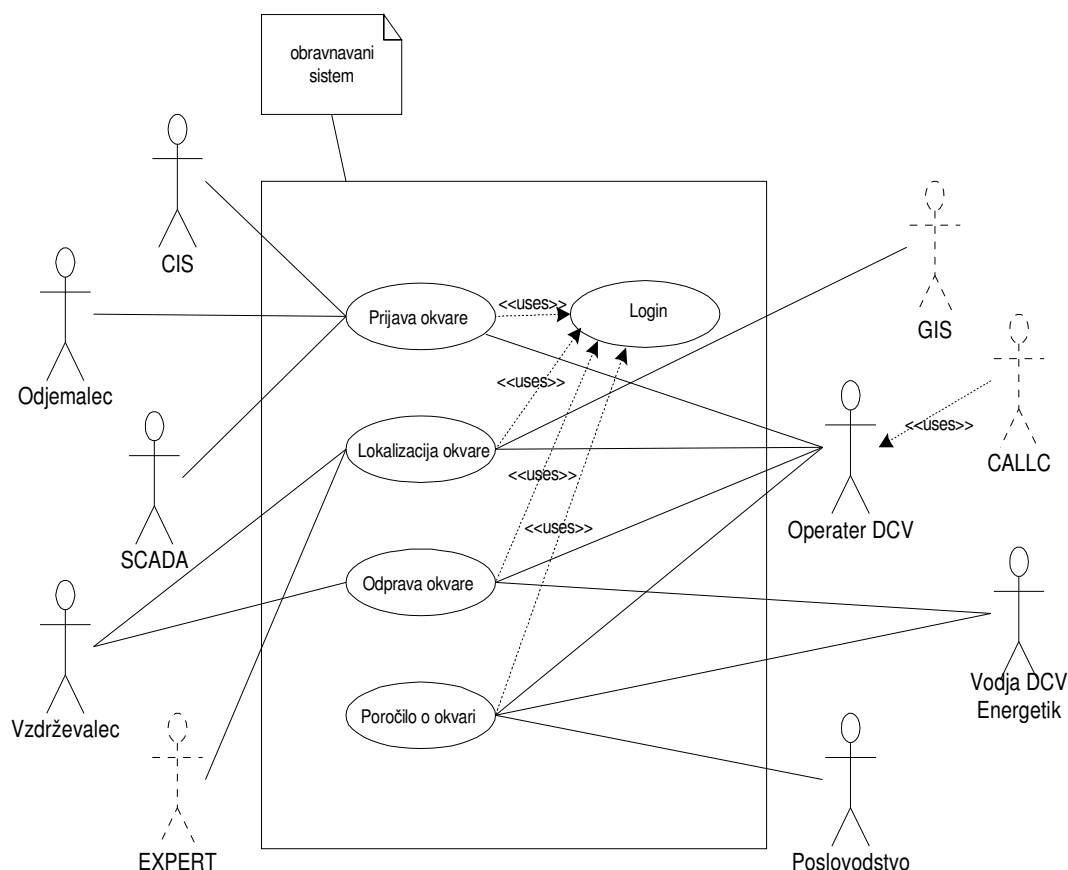
1. Prijava okvare: aktivnosti, ki se nanašajo na sprejem in evidentiranje klicev odjemalcev, evidentiranje okvare in dajanje informacij v zvezi z okvaro, generiranje ad hoc informacij o številu prizadetih odjemalcev, evidentiranje občutljivih odjemalcev, izračun nedobavljene električne energije itd.
2. Lokalizacija okvare: aktivnosti, ki so povezane z lokalizacijo okvarnega dela omrežja in vključujejo korelacijo klicev in signalizacije izpada omrežja iz SCADA sistema ter vodenje stikalnih postopkov za izolacijo dela omrežja, ki je v okvari. V nadgradnji sistema bo vključeno tudi predlaganje postopkov za lokalizacijo okvare s pomočjo ekspertnega sistema.

3. Odprava okvare: aktivnosti, ki so vezane na pripravo dela, odpravo okvare ter vzpostavitev normalnega stanja omrežja.
4. Obveščanje in priprava poročil: funkcionalnost sistema, ki omogoča ažurno sledenje postopkov del in pripravo obvestil in poročil poslovodstvu ter zunanjim akterjem.

Slika 40: Akterji v obravnavanem sistemu



Slika 41: Model sistema s temeljnimi primeri uporabe



Primer uporabe »Prijava okvare«

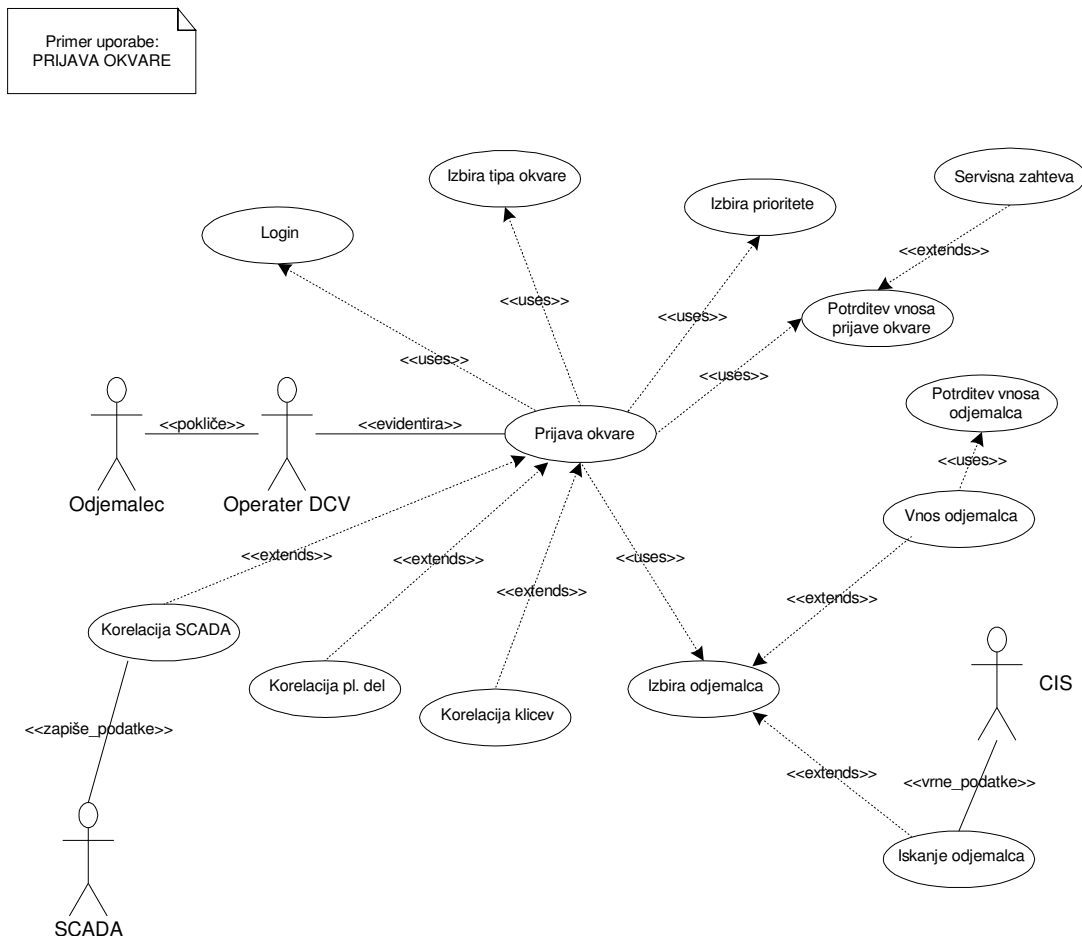
Primer uporabe *Prijava okvare* opredelimo kot funkcionalnost sistema, ki vključuje prijavo okvare na podlagi telefonskega klica odjemalca, evidentiranje klica odjemalca, korelacijo okvare s predhodnimi klici odjemalcev, s signaliziranim izpadom dela omrežja iz SCADA sistema in s tekočimi planskimi deli ter obveščanje odjemalca o okvari in postopkih v zvezi z njeno odpravo.

Postopki znotraj scenarija primera uporabe *Prijava okvare* so sledeči:

1. Odjemalec pokliče operaterja v DCV. Operater sprejme klic.
2. Operater zažene aplikacijo in se prijavi v sistem (*Login*).
3. Operater na podlagi podatkov, ki jih pridobi od kličočega, izbere s seznama odjemalcev odgovarjajočega. Iz interne baze podatkov se v polja okenske maske prepisejo podatki, vezani na izbranega odjemalca.
4. Če sistem ne najde kličočega znotraj svoje baze podatkov, izbere operater možnost iskanja odjemalca v zunanji podatkovni bazi CIS. Generira se poizvedba po odjemalcu glede na različne parametre. Če poizvedba najde pravi zadetek, se podatki o odjemalcu prepisejo v ustrezna polja vnosne maske. Hkrati se podatki o odjemalcu shranijo v interno bazo podatkov sistema.
5. Če tudi zunanji informacijski sistem CIS ne najde odgovarjajočega odjemalca, potem operater ročno vnese zahtevane podatke, ki se shranijo v interno bazo podatkov sistema. Ažuriranje zunanje baze podatkov je stvar skrbnikov CIS.

6. Operater na podlagi informacij, ki jih pridobi od klicočega odjemalca, vnese tip okvare pri odjemalcu in njeno prioriteto obravnave.
7. Operater potrdi vnos prijave okvare, kjer se zabeležijo podatki o klicu in okvari. Dodatno ima operater možnost generiranja servisne zahteve, ki je vezana na klicočega odjemalca. Izdaja servisne zahteve je predvsem pomembna za primere, ko odjemalec ali več odjemalcev prijavi okvaro, ki ni zaznana v SCADA sistemu.
8. Operater nudi odjemalcu dodatne informacije, ki jih pridobi na podlagi izvedbe korelacije z izpadom omrežja iz SCADA sistema, na podlagi klicev predhodnih klicev odjemalcev in na podlagi evidentiranih planskih del.

Slika 42: Primer uporabe *Prijava okvare*

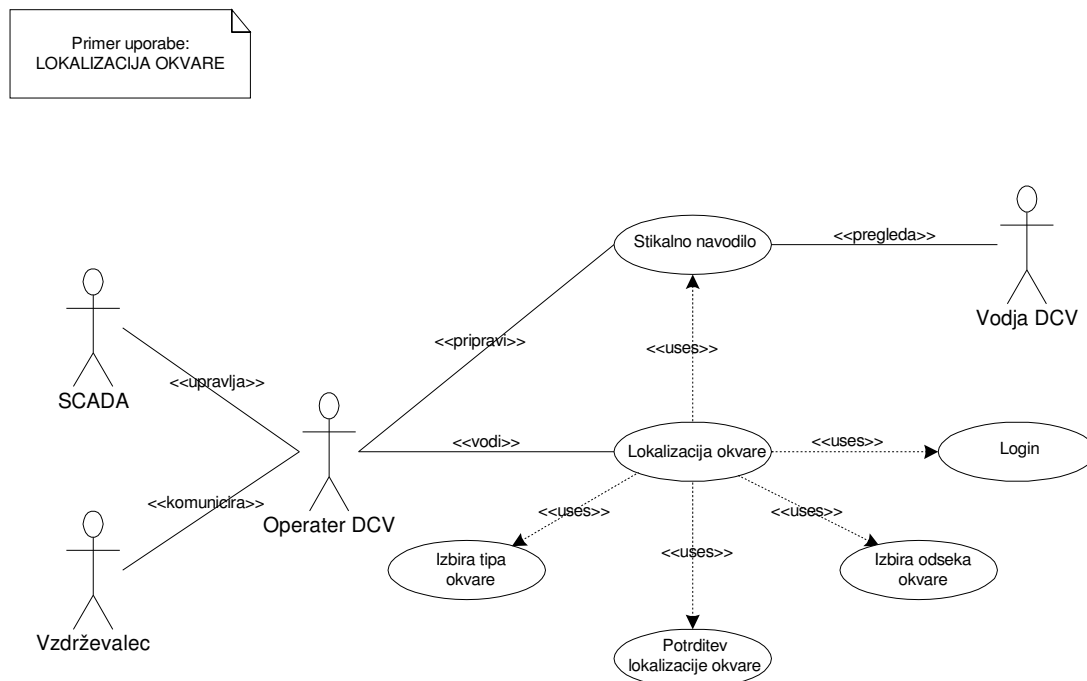


Primer uporabe »Lokalizacija okvare«

Primer uporabe *Lokalizacija okvare* opredelimo kot funkcionalnost sistema, ki podpira postopke za čimbolj natančno določitev mesta in tipa okvare določenega dela omrežja. Na podlagi signalizacije izpada dela omrežja iz SCADA sistema in na podlagi korelacije klicev odjemalcev lahko z omejeno natančnostjo opredelimo mesto okvare omrežja. Funkcionalnost tega primera uporabe opišemo na sledeč način:

1. Operater določi okvirno področje okvare omrežja, to je navadno srednjenapetostni izvod omrežja (preko SCADA sistema) ali odsek izvoda omrežja ali transformatorska postaja (preko korelacije klicev).
2. Operater pripravi stikalno navodilo z vsemi potrebnimi stikalnimi manipulacijami za lokalizacijo okvarnega mesta.
3. Vodja DCV oziroma vodja izmene pregleda navodilo. Tu gre za pregled tehnično – operativne ustreznosti, zato ta postopek ni formalen v smislu odobritve po pravilih varstva pri delu. Ker gre samo za izvajanje stikalnih manipulacij, ki se lahko opravljajo tudi zunaj rednega delovnega časa, se za ta primer ne zahtevajo posebni varnostni ukrepi.
4. Če ima vzdrževalec na terenu možnost dostopa do interneta, si lahko pripravljeno navodilo stiska, sicer mu operater navodila posreduje preko UKV zveze.
5. Operater vodi postopek lokalizacije okvare tako, da upravlja s SCADA sistemom (daljinsko krmili stikala, ki so daljinsko nadzorovana) in z vzdrževalcem na terenu (posreduje navodilo za izvedbo posamezne stikalne manipulacije).
6. Ko sta lokacija in tip okvare določena in je okvarjeni odsek izoliran od neprizadetega dela omrežja, operater potrdi lokalizacijo okvare. Sledijo postopki, ki se nanašajo na primer uporabe *Odprava okvare*.

Slika 43: Primer uporabe *Lokalizacija okvare*

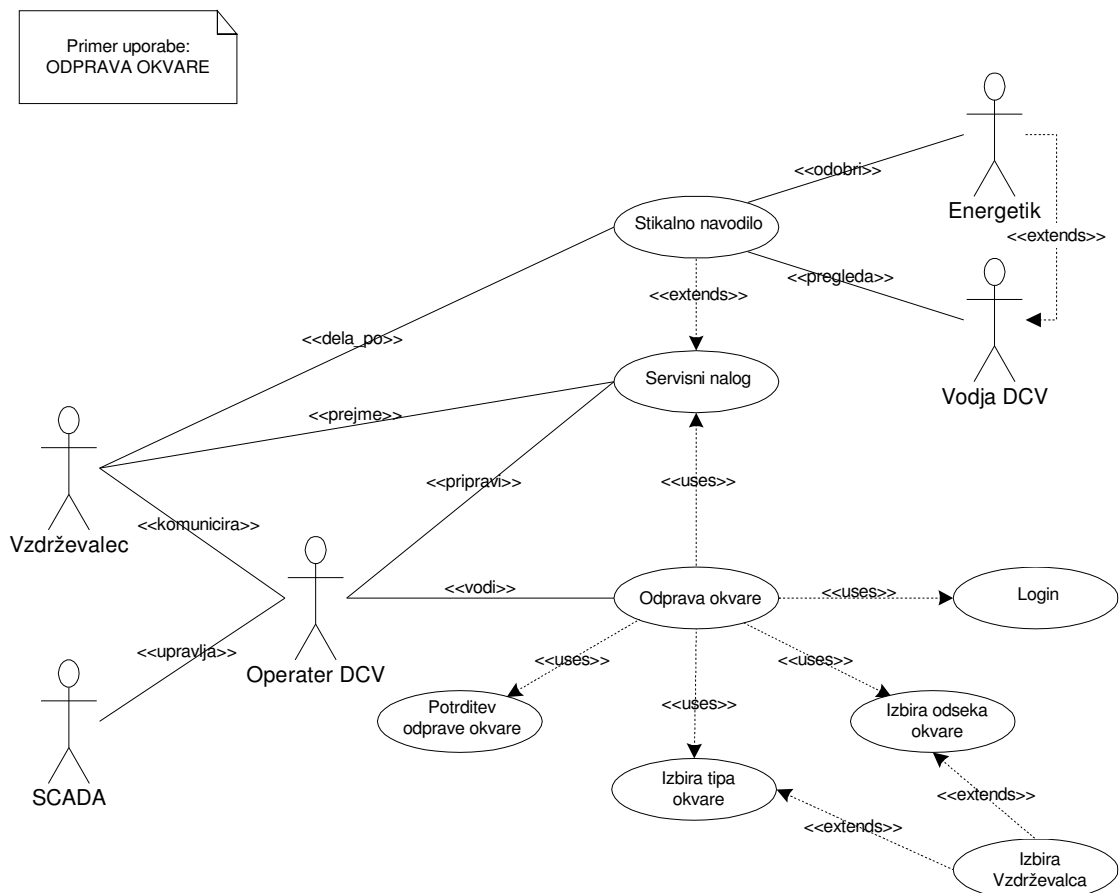


Primer uporabe »Odprava okvare«

Primer uporabe *Odprava okvare* opredelimo kot funkcionalnost sistema, ki podpira postopke za odpravo okvare na DEES. Primer uporabe *odprava okvare* se nanaša tako na odpravljanje okvare pri odjemalcu (npr. pregorela varovalka) kot tudi na odpravljanje okvare na omrežju, ko je prizadetih več odjemalcev. Funkcionalnost tega primera uporabe opišemo na sledeč način:

1. Operater na podlagi lociranega mesta okvare pripravi servisni nalog. Če gre za odpravo okvare na omrežju, potem razširi funkcionalnost s pripravo stikalnega navodila za odpravo okvare in izvedbo energetskega prenapajanja omrežja. Pri odpravi okvare lahko sodeluje več vzdrževalnih ekip, zato takrat pripravi več servisnih nalogov.
2. Vodja DCV oziroma vodja izmene pregleda navodilo in ga da v odobritev Energetiku.
3. Servisni nalog je posredovan Vzdrževalcu, ki je zadolžen za odpravo okvare pri odjemalcu oziroma za odpravo okvare na omrežju. V drugem primeru prejme tudi odobreno stikalno navodilo, ki mu nalaga izvedbo potrebnih manipulacij in ukrepov iz varstva pri delu. Šele nato se lahko loti ustreznih popravil.
4. Ko je okvara odpravljena, vzdrževalec javi Operaterju, da so dela zaključena.
5. Če je šlo za odpravo okvare pri odjemalcu, Operater zaključi servisni nalog in tako potrdi odprave okvare. Hkrati se zaključi tudi servisna zahteva, ki je bila generirana na podlagi klica odjemlaca.
6. Če je šlo za odpravo okvare na omrežju, Operater in Vzdrževalec izvedeta potrebne stikalne manipulacije, da se vzpostavi prvotno stanje omrežja in izvede napajanje vseh odjemalcev. Operater zaključi servisni nalog oziroma servisne naloge. Ko so vsi nalogi zaključeni, potrdi odpravo okvare in hkrati se zaključijo tudi servisne zahteve, vezane na predmetno okvaro.

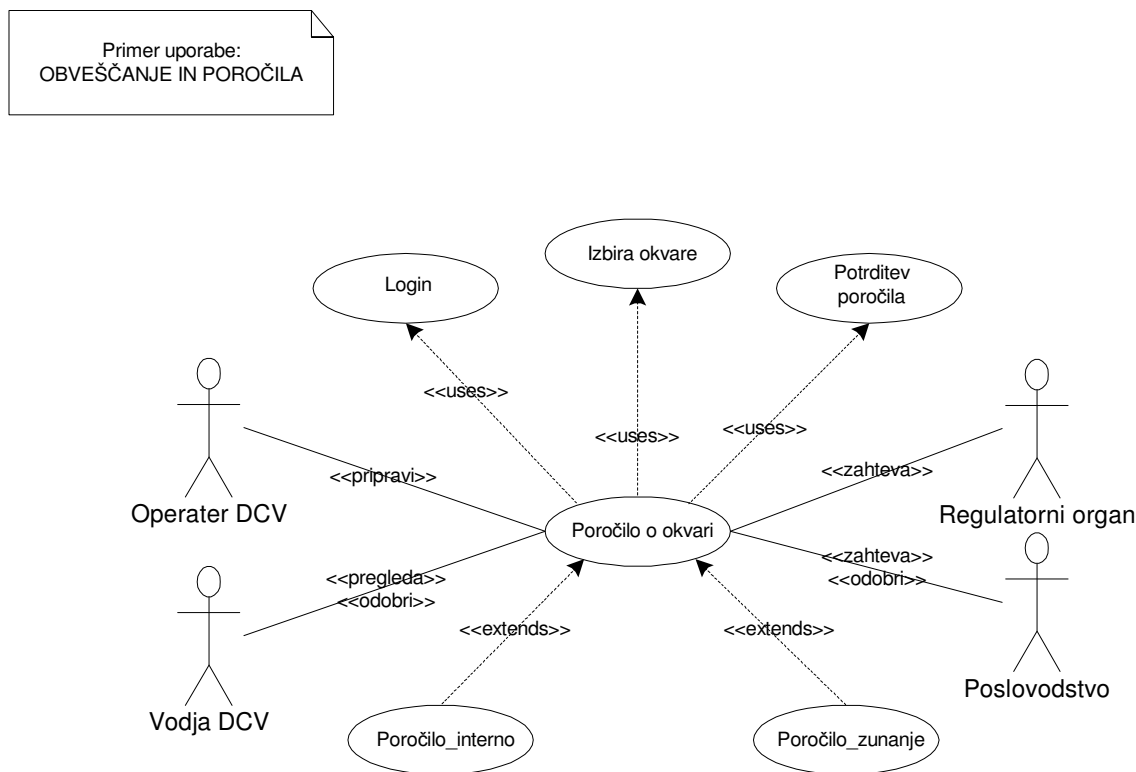
Slika 44: Primer uporabe *Odprava okvare*



Primer uporabe »Obveščanje in priprava poročil«

Primer uporabe *Obveščanje in priprava poročil* opredelimo kot funkcionalnost sistema, ki omogoča oblikovanje in pripravo različnih poročil za notranje (poslovodstvo) in zunanje uporabnike (regulatorni organi, mediji itd.). Hkrati omogoča tudi izvajanje različnih poizvedb po okvarah in prijavah ter pridobivanje parametrov in kazalnikov o okvarah. Na sliki prikazana funkcionalnost bo lahko nadgrajena z izračunom nedobavljene energije, s številom odjemalcev brez napajanja ipd.

Slika 45: Primer uporabe *Obveščanje in poročila*



5.3.2.2 Diagrami statične strukture sistema

Kot omenjeno, predstavljajo diagrami statične strukture sistema diagrame razredov in objektov ter relacij med njimi. Diagrami objektov so implementacije diagramov razredov in prikazujejo primerke, ki so skladni z diagrami razredov. Razredi določajo skupne značilnosti (atribute) objektov in njihovo obnašanje (operacije, metode).

Iz problemskega dokumenta in diagramov primerov uporabe identificiramo naslednje kandidate za razrede:

Slika 46: Kandidati za razrede

Kandidati za razrede				
Operater_DCV	Vodja_DCV	Energetik	Poslovodstvo	Vzdrževalec
Okvara	Odjemalec	Reg_organ	Tel_klic	Prijava_okvare
Odprava_okvare	Lokalizacija_okvare	Vodja_nadzorništva	Poročilo_okvara	Zahteva_poročilo
Izvod_omrežja	Odsek_omrežja	Izpad	Trafo_postaja	EEnaprava
Servisni_nalog	Servisna_zahteva	Stikalno_navodilo	Plan_dela	Delovni_nalog
Potrditev_prijave	Vzdrz_ekipa	Dovoljenje_za_delo	Izdaja	Popravilo
Servis	Lokacija_okvare	Delovno_mesto	Zaposleni	Potrditev_okvare
Kontakt_naslov	Poslovni_partner	Login		

Po preverjanju definicije kandidatov za razrede ugotovimo:

- ❑ Razredi *Operater_DCV*, *Vodja_DCV*, *Energetik*, *Poslovodstvo*, *Vzdrževalec*, *Vodja nadzorništva* opisujejo entitete, ki imajo skupno obnašanje kot ga ima v asociacijah statičnega modela razred **Zaposleni**.
- ❑ Razreda *Reg_organ* in *Odjemalec* sta podrazreda razreda *Posl_partner*.
- ❑ Razredi **Okvara** in *Prijava_okvare* sta redundantna razreda.
- ❑ Razreda **Popravilo** in *Servis* sta redundantna razreda.
- ❑ Razred *Lokacija_okvare* je atribut razreda *Okvara*.
- ❑ Razreda **Servisni_nalog** in *Delovni_nalog* sta sinonima in redundantna.
- ❑ Razredi *Lokalizacija_okvare*, *Odprava_okvare*, *Potrditev_okvare*, *Tel_klic*, *Potrditev_prijave*, *Izdaja* so nejasni in jih eliminiramo.
- ❑ Razred *Dovoljenje_za_delo* je dokument, ki si ga izmenjajo odgovorni upravljalci objekta in vzdrževalci in zato nima neposredne zveze z obravnavanim sistemom.
- ❑ Razred *Login* je implementacijski konstrukt in v tej fazi analize problema ne pride v poštev za obravnavo.

V statični objektni model tako vključimo naslednje razrede:

Zaposleni

So vsi delavci podjetja za distribucijo električne energije. Objekti tega razreda imajo različne vloge v obravnavanem modeliranem sistemu.

Okvara

Okvara pri odjemalcu ali na omrežju. Vsebuje vse podatke povezane z okvaro.

Poslovni_partner

Nadrazred, ki opredeljuje skupne lastnosti odjemalcev in ostalih partnerjev.

Odjemalec

Razred, ki določa fizično ali pravno osebo, priključeno na elektroenergetsko omrežje, ki nima možnosti izbire dobavitelja in je podvržena tarifnemu sistemu, ki ga določi vlada RS.

Reg_organ

Razred, ki opredeljuje regulatorni organ, ki bdi nad izvajanjem javnih služb v podjetju (agencija, inšpektorat, ministrstvo). Organ zahteva ustrezna poročila o dogodkih v omrežju.

Poročilo_okvara

Dokument, ki se dostavi na zahtevo in vsebuje podatke vezane na okvaro.

Zahteva_poročilo

Dokument, ki se posreduje v DCV v smislu priprave ustreznega poročila.

Servisni_nalog

Dokument, ki je dostavljen vzdrževalcu za odpravo okvare oziroma za vzdrževalna dela.

Servisna_zahteva

Dokument, ki se generira kot posledica prijave okvare s strani odjemalca.

Stikalno_navodilo

Natančen postopek izvajanja stikalnih manipulacij.

Plan_dela

Dokument, kjer so zabeležena planirana dela za posamezne dneve v letu.

Popravilo

Se nanaša na popravilo EE naprave, ki je bila vezana na določeno okvaro.

EEnaprava

Elektroenergetska naprava, ki je predmet oziroma sredstvo popravila ali vzdrževanja.

Vzdrževalna_ekipa

Ekipa na pristojnem območju, zadolžena za odpravo posameznih vrst okvar.

Izvod_omrežja

Elektroenergetski vod, ki napaja določeno področje s transformatorskimi postajami.

Odsek_omrežja

Najmanjši del izvoda omrežja, ki ga je mogoče s stikalnimi manipulacijami ločiti od ostalega dela izvoda omrežja.

Trafo_postaja

Napajalna postaja na SN omrežju, ki pretvarja srednjo na nizko napetost (NN) in napaja pripadajoče odjemalce.

Izpad

Izpad izvoda omrežja pomeni prekinitev napajanja trafo postaj in odjemalcev, tesno povezan z izpadom, ki ga signalizira SCADA sistem.

Delovno_mesto

V sistemizaciji delovnih mest opisano delovno mesto, na katerem dela določeni zaposleni v podjetju.

Kontakt_naslov

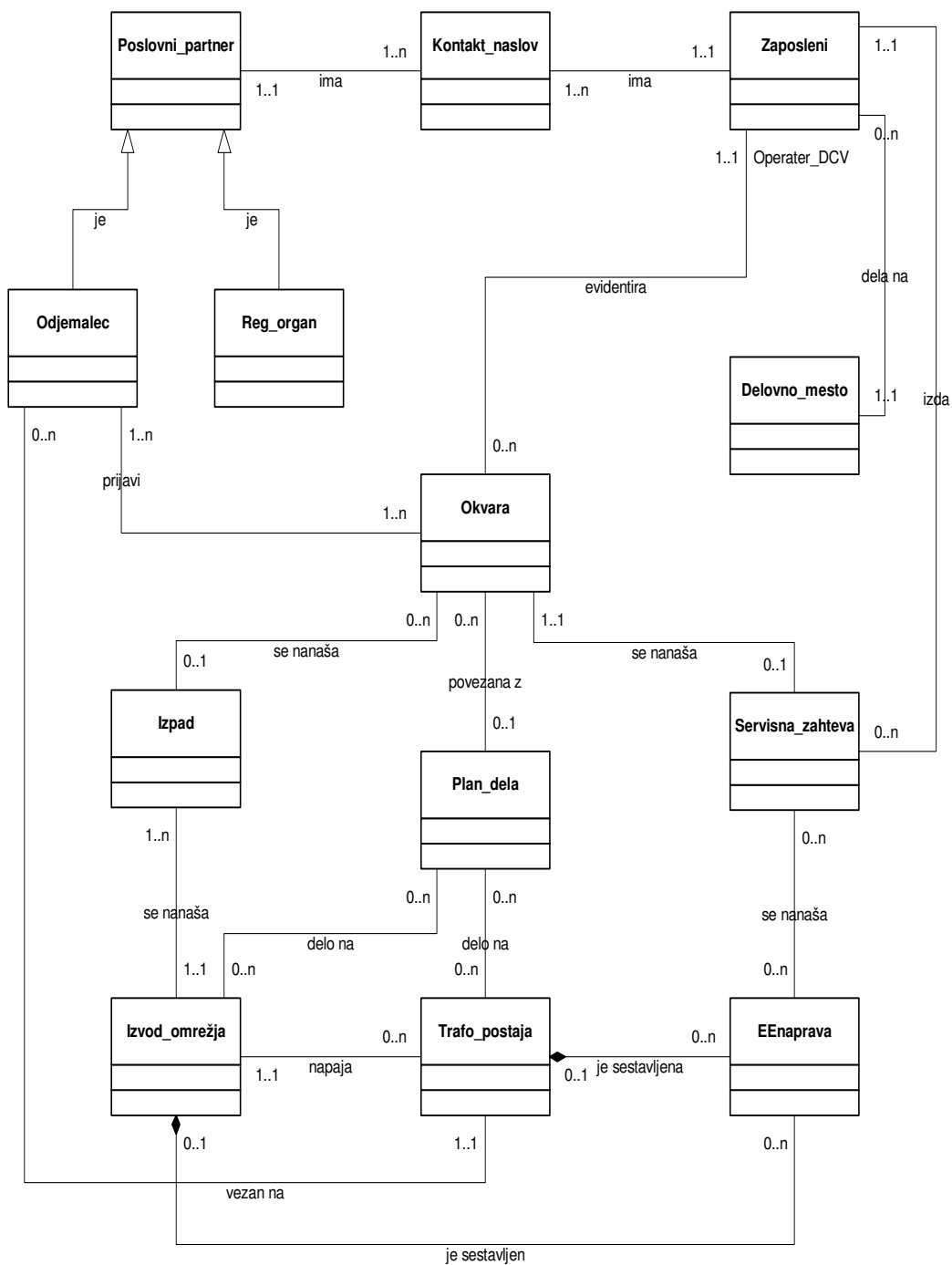
Vsebuje osebne in kontaktne podatke, ki so vezani na zaposlene in poslovne partnerje.

Rezultat prve obdelave razredov je začetni objektni model s prečiščenimi razredi brez atributov. Na sliki 47 je predstavljen prvi del začetnega diagrama razredov, nadaljevanje diagrama pa je prikazano v prilogi tega dela. Nekateri razredi na slikah zaradi preglednosti nastopajo večkrat.

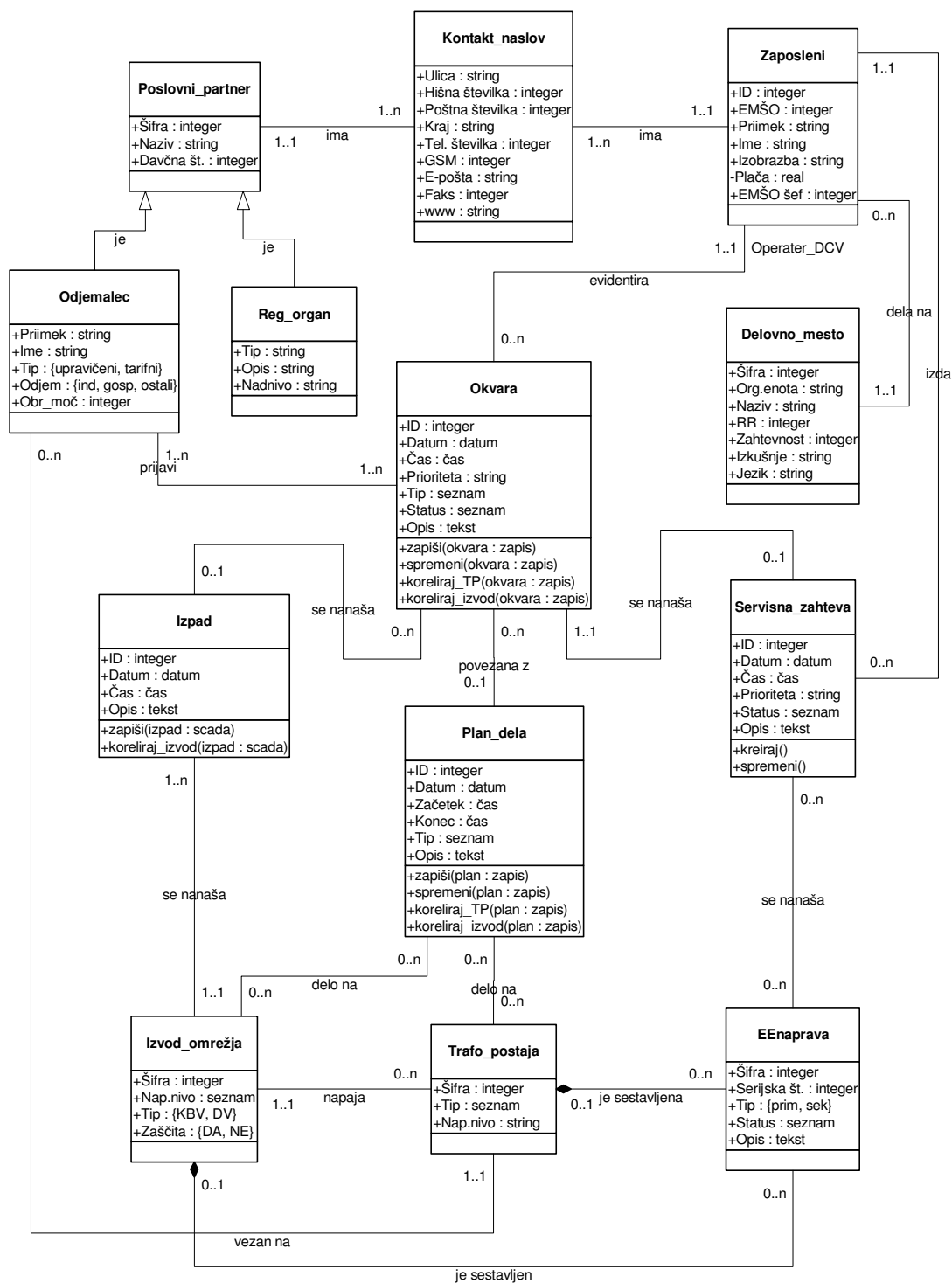
Pri rekurzivnem pregledovanju in analizi statičnega modela ugotavljam nekatere pomanjkljivosti začetnega razrednega modela. Nekatere popravke sem vnašal sproti, tako da tu navajam samo to, da sem nekaterim asociacijam popravil kardinalnost, razred *Poročilo_okvara* pa sem na novo opredelil kot nadrazred in njemu priredil nova podrazreda *Poročilo_zunanje* in *Poročilo_notranje*. V končnem modelu sem tudi predvidel, da lahko poročilo o okvari zahteva tudi *Odjemalec*. Iz začetnega modela lahko tudi eliminiram asociacijo »zahteva« med razredoma *Reg_organ* in *Poročilo_okvara* ter med razredoma *Zaposleni (vloga Poslovodstvo)* in *Poročilo_okvara*. Obe asociaciji je namreč mogoče izpeljati iz asociacij »izda« napram razredu *Zahteva_poročilo*.

V končnem modelu so identificirani atributi razredov in ključne operacije sistemskih razredov. Zaradi preglednosti modela nisem navedel vseh operacij za vse razrede, ker bi to sicer prekoračilo zastavljen obseg dela. Na sliki 48 je predstavljen prvi del končnega diagrama razredov, nadaljevanje diagrama pa je prikazano v prilogi tega dela.

Slika 47: Začetni diagram razredov



Slika 48: Končni diagram razredov



5.3.2.3 Diagrami dinamičnega modela sistema

Pogledi na sistem, ki se nanašajo na časovno dimenzijo objektov in spremembe, ki so jim le-ti podvrženi, sestavljajo dinamični model obravnavanega sistema. Gre za opis zaporedja operacij, ki se pojavijo kot odgovor na zunanje stimulacije oziroma dogodke. Gre za opis dinamičnega obnašanja objektov in sistema v celoti. Objekti sistema interagirajo med seboj tako, da si izmenjujejo informacije. S tem povzročajo spremembe svojih stanj oziroma trenutnih vrednosti svojih atributov.

Dinamično obnašanje objektov in sistema opišemo z diagrami stanj, diagrami aktivnosti, zaporednostnimi diagrami in diagrami sodelovanja. V nadaljevanju so dinamično obdelani prvi trije naštetih pogledi na sistem. Diagrami sodelovanja prikazujejo interakcijo med objekti in so v smislu pogleda na sistem za časovno komponento interakcij okrnjeni zaporednostni diagrami, zato v nadaljevanju ne bodo predstavljeni.

Diagrami stanj

Diagram stanj prikazuje življenjski cikel določenega objekta. Prikazuje dogodke, ki so povzročili prehode iz enega stanja v drugega, in aktivnosti, ki izhajajo iz spremembe stanja objekta. Dogodki so lahko sporočila, zahteve, klici procedur, posegi akterjev itd. V določeni meri je iz njega možno izpeljati tudi diagram aktivnosti, ki bo obravnavan v nadaljevanju. Predvsem na tak način obdelamo tiste razrede objektov, ki imajo tipično veliko dinamičnega obnašanja.

V nadaljevanju bom obdelal razreda *Okvara* in *Stikalno_navodilo*, ki je v povezavi z razredom *Servisni_nalog*. Razred *Okvara* je osrednji razred obravnavanega sistema. Vse dogajanje se vrti okrog prijave, zaznave, lokalizacije in odprave okvare. V sklopu obravnave okvare je zelo pomemben razred *Stikalno_navodilo*, ki je vključen v njeno lokalizacijo in odpravo.

Dejavnost določenega objekta v nekem stanju imenujemo njegovo aktivnost. Aktivnost razlikujemo od pojma »akcija«, ki je bolj povezan s prehodi med posameznimi stanji in predstavlja prožilne dogodke, ki te prehode povzročijo. Za akcije opredelimo dogodke tipa *entry*, *exit*, *do* ipd.

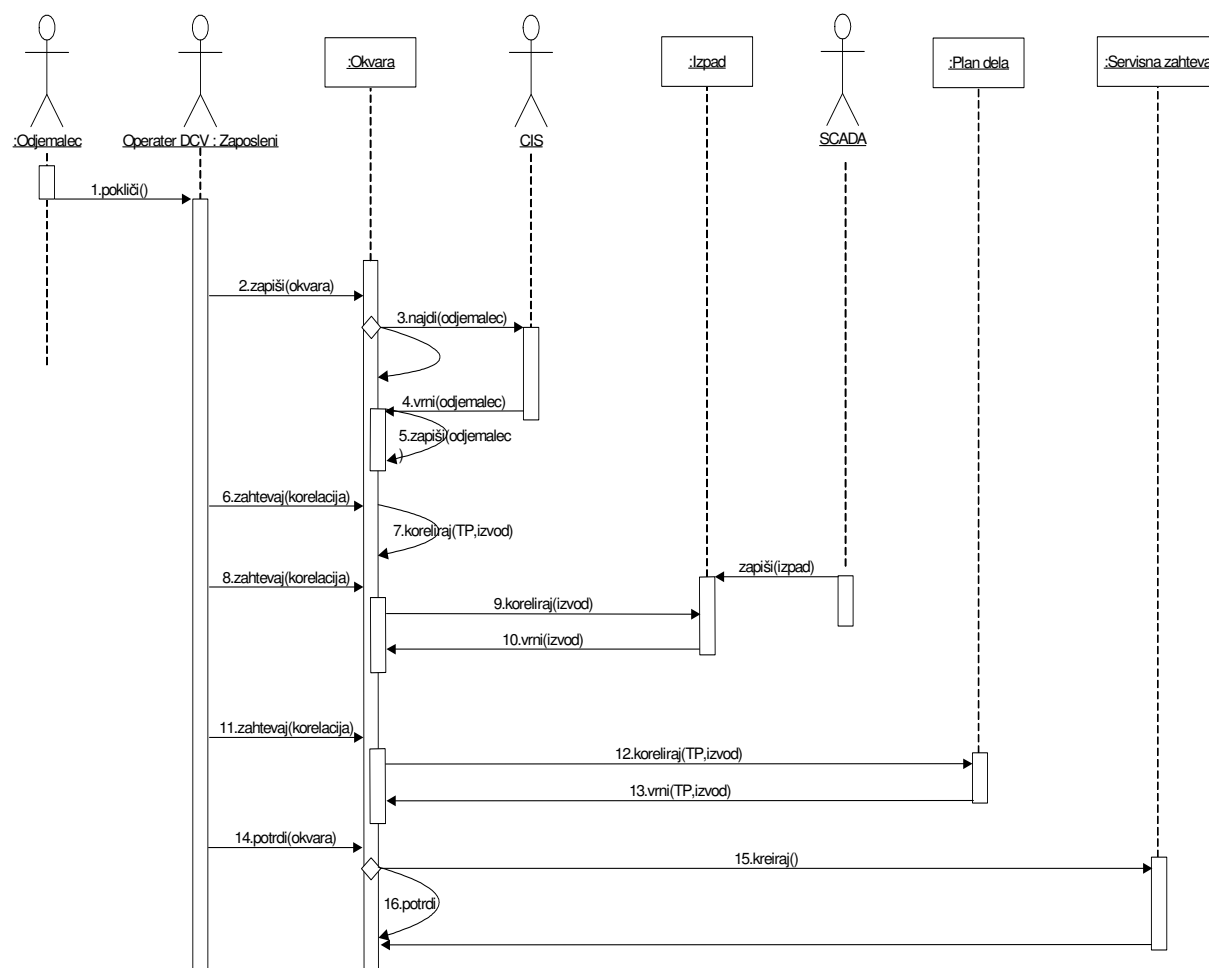
V prilogi tega dela sta prikazana diagrama stanj za omenjena razreda in sta povezana z opisom v problemskem dokumentu.

Zaporednostni diagrami

Zaporednostni diagrami prikazujejo, kako objekti medsebojno interagirajo. Osredotočajo se na zaporednost sporočil, ki se pošiljajo med objekti in na tak način prožijo določene aktivnosti objektov v določenih stanjih. Zaporednostni diagrami podrobno izkazujejo interakcije med objekti za nek določen scenarij v časovnem zaporedju. Zelo podobni za predstavitev sodelovanja med objekti so diagrami sodelovanja, ki izkazujejo le drugačen pogled na scenarij, ki ga prikazujejo zaporednostni diagrami, ne izkazujejo pa časovne dimenzije interagiranja med objekti. V literaturi tudi najdemo izraz diagrami interakcij, pod katerim razumemo tako zaporednostne diagrame kot tudi diagrame sodelovanja.

Na sliki 49 je s pomočjo zaporednostnega diagrama predstavljen scenarij za primer uporabe *Prijava okvare*, ostala dva scenarija sta prikazana v prilogi tega dela.

Slika 49: Zaporednostni diagram *Prijava okvare*

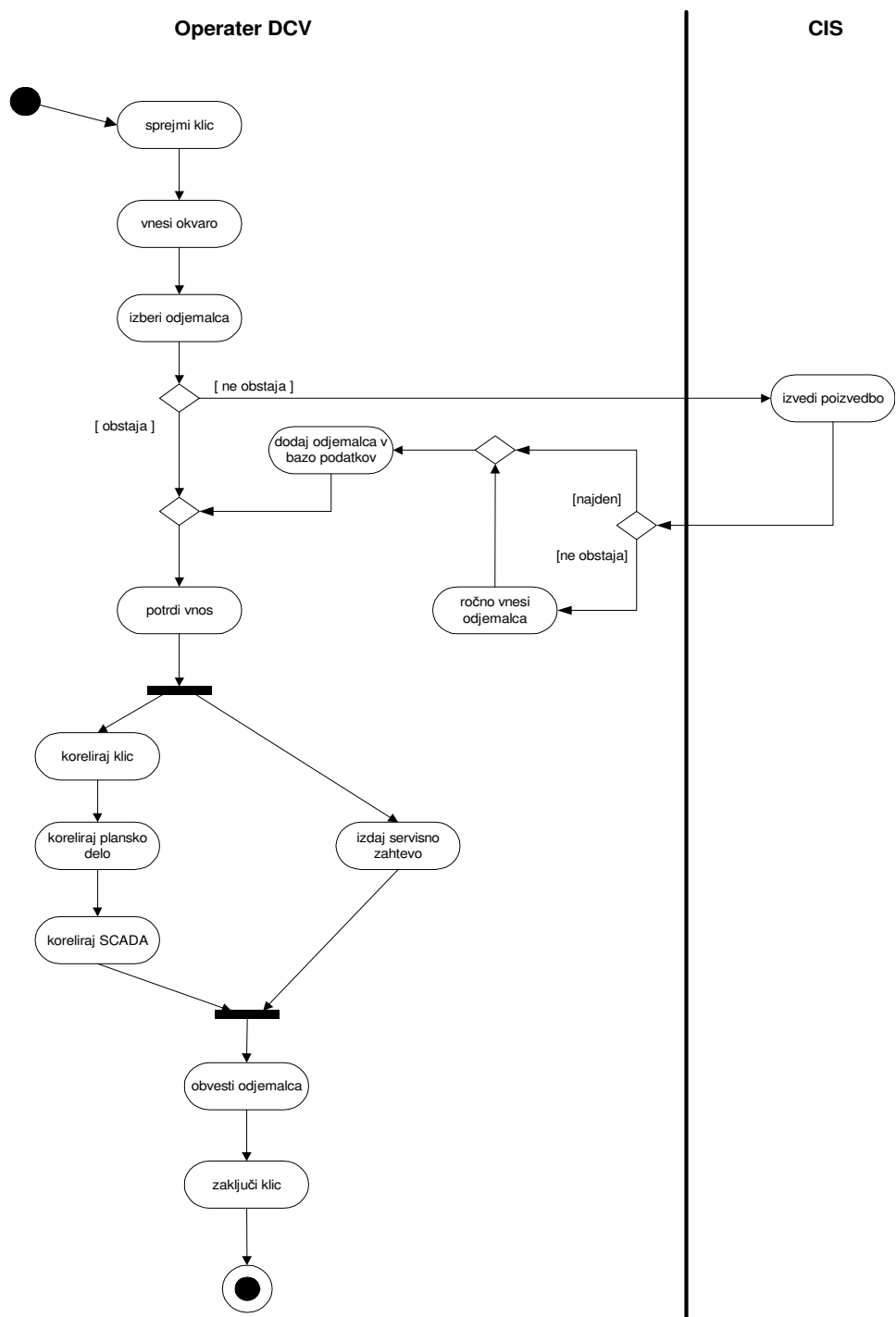


Diagrami aktivnosti

Diagrami aktivnosti se osredotočajo na izvajanje posameznih dejavnosti znotraj implementacije določene operacije (metode), lahko pa prikazujejo tudi izvajanje aktivnosti v sklopu določenega primera uporabe ali znotraj objektov samih.

Na sliki 50 je predstavljen diagram aktivnosti za primer uporabe *Prijava okvare*. Diagrama aktivnosti za primera uporabe *Lokalizacija okvare* in *Odprava okvare* sta prikazana v prilogi tega dela.

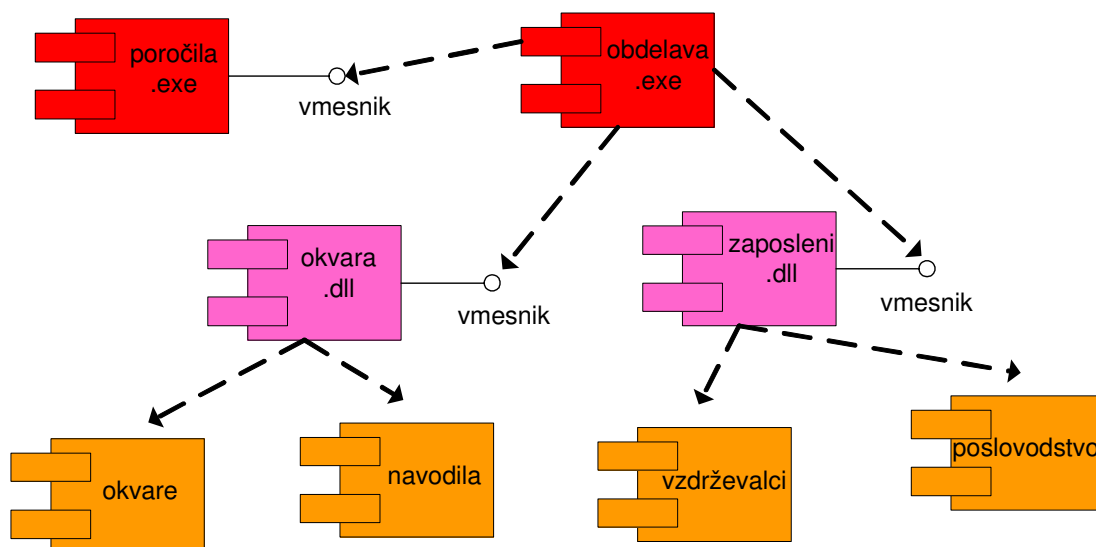
Slika 50: Diagram aktivnosti za primer *Prijava okvare*



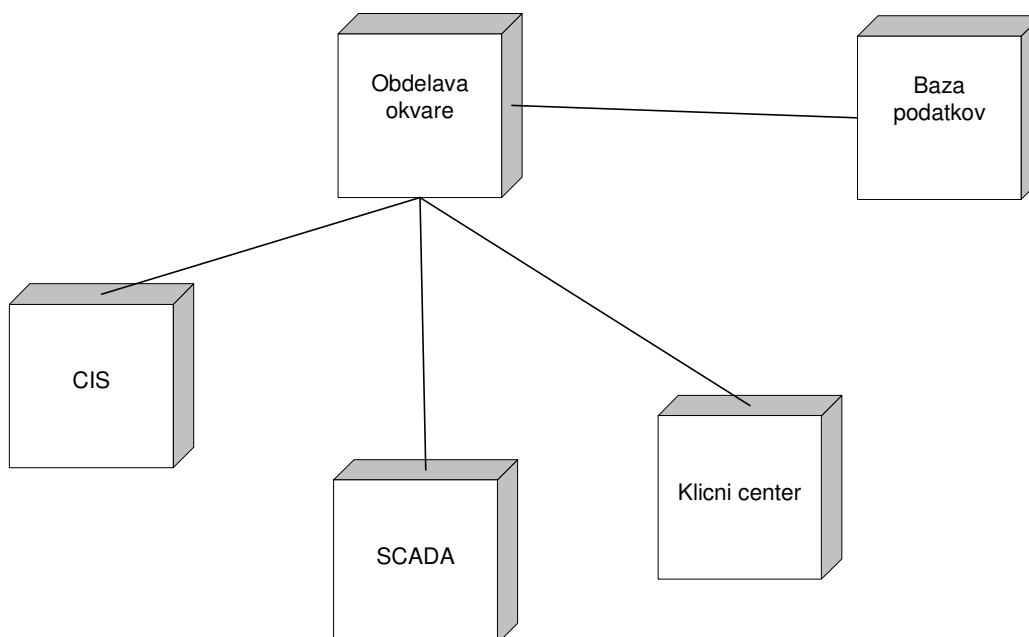
5.3.2.4 Diagrami implementacije

Fazi analize in načrtovanja sistema sledi faza implementacije, ki se sprti preverja in testira. UML notacija ponuja dva tipa tovrstnih diagramov, in sicer diagram komponent⁵⁸ in diagram porazdelitve⁵⁹. Prvi prikazuje odvisnosti med posameznimi programskimi komponentami (izvorna koda, binarna koda, izvršilna koda) in strukturo kode (slika 51), drugi pa prikazuje konfiguracijo izvršilnih procesnih elementov ter programskih komponent, procesov in objektov, ki se na teh procesnih elementih izvajajo (slika 52).

Slika 51: Diagram komponent obravnavanega podsistema



Slika 52: Diagram porazdelitve obravnavanega podsistema



⁵⁸ Component diagram

⁵⁹ Deployment diagram

6 SKLEP

Učinkovito in visoko kvalitetno načrtovanje sistemov vodenja in avtomatizacije je velik izziv že od samega začetka uvajanja takih sistemov v procese obvladovanja elektroenergetskega sistema. Učinkoviti pristopi k specifikaciji in načrtovanju procesnih podatkovnih baz in sistemov so lahko odločilnega pomena za doseganje želenega cilja. Bistvena pri tem je zato, v sklopu informatizacije procesov, uporaba ustreznih modelirnih tehnik.

Opisano aparaturno modeliranje podatkov in procesov, ki skupaj tvorijo procesni sistem, ponuja modularne rešitve, ki omogočajo učinkovito, racionalno in odprto gradnjo procesnih podatkovnih baz in sistemov ter v bistvu sovpadajo z objektnim modeliranjem sistemov realnega sveta. Aparaturno-objektno modeliranje sistemov je modularno in odprto za načrtovanje poljubnega števila EEP. Temelji na uniformni procesni podatkovni bazi, ki jo je mogoče učinkovito dopolnjevati glede na spremembe na primarni fizični procesni opremi. Objekti kot odslikave realnega sveta imajo svojo lastno identiteto, kar omogoča enostavno odstranjevanje obstoječih objektov, predvsem pa omogoča dodajanje novih na način, da se lahko posledice tovrstnih sprememb lokalizirajo na objektih, ki so v interakciji s spremenjenimi objekti. Tako je možnost negativnih stranskih učinkov zmanjšana na minimum in tako je tudi zagotovljena konsistentnost delovanja sistema. Omenjeno omogočajo predvsem objektni koncepti razrednega dedovanja, tako v času izgradnje (analiza in načrtovanje sistema) in nadgradnje sistema (dodajanje novih objektov) kot tudi za potrebe analiz dogajanj v primarnem sistemu. In to tudi tedaj, ko ni mogoče neposredno dedovanje lastnosti neke že opisane naprave realnega sveta (razreda). Takrat je možno že zgrajeni razred ponovno uporabiti in dodati le lastnosti, ki jih le-ta ni zaobjel.

V nasprotju s tem ima v tej nalogi predstavljeno klasično funkcijsko modeliranje nekatere specifične omejitve. Tako zgrajeni modeli so namreč strogo namenjeni za konkretno aplikacijo v nekem EEP in ne omogočajo modularne in odprte nadgradnje za potrebe širše uporabe tudi na drugih konkretnih primerih. So tudi odvisni od uporabljenih tehnoloških rešitev. Naprave nimajo nobene povezave niti med sabo niti s podatki, ki jih bodisi generirajo bodisi sprejemajo. Zato v procesu načrtovanja prihaja do nepotrebnih podvojevanj vnosov.

Sistemi grajeni na osnovi objektno podstati izkazujejo prednosti ne samo v fazi modeliranja, izgradnje, nadgradnje in vzdrževanja, ampak tudi takrat, ko je potrebno izvajati analizo dogajanj v operativnih procesih in analizo dogodkov znotraj primarnih sistemov, ki jih informacijsko obvladujemo. Z naraščanjem obsežnosti in kompleksnosti sistema so analize temelječe na sprotnih izpisih nepovezanih dogodkov zahtevne in neučinkovite. Vsebinsko povezani dogodki so namreč poljubno izpisani in ni možnosti njihove medsebojne korelacije, ki sicer v realnem okolju obstaja. Vzrok temu je v njihovem modeliranju, to je že v zgodnji fazi načrtovanja sistema. Signali iz realnega sistema so obravnavani posamezno glede na njihovo funkcijo. Seveda obstaja tudi cena za učinkovito upravljanje kompleksnosti sistema, saj je nujnost, poleg neposrednih investicijskih stroškov uvedbe novega sistema, pristopiti k formalnemu opisu vsakega, na prvi pogled še tako nebistvenega dela primarnega sistema. Vložen začetni trud se povrne predvsem takrat, ko je mogoče zgrajen sistem učinkovito in enostavno razširjati z uporabo že zgrajenih objektov. In ne nazadnje, ko se centralni sistemi

vodenja, zgrajeni na klasični funkcijski podstati, nezadržno širijo v obsegu in kompleksnosti, postaja tudi vzdrževanje njihovih programskih sklopov praktično neobvladljivo.

Velja tudi omeniti, da se v strokovnih krogih mednarodnega združenja IEC pripravljajo priporočila oblikovana v nastajajočih IEC standardih, in sicer IEC 61850, 61970 in 61968, ki se soočajo z obravnavano problemsko domeno. IEC 61850 se ukvarja s komunikacijskimi omrežji in sistemi ter z zahtevami za njihovo aplikacijo in modeliranjem v okviru posameznega EEP. Posega zelo globoko v zgradbo naprav. IEC 61970 se ukvarja z aplikacijskimi programskimi vmesniki⁶⁰ za upravljanje omrežja, z modeliranjem enotnega informacijskega modela⁶¹ EES (t.i. CIM) in povezljivostjo aplikacij znotraj ožjega centra vodenja. Osnovni namen izdelave CIM je bil razviti podporo integraciji aplikacij različnih proizvajalcev v enoten sistem za upravljanje omrežja (Podmore, 1999, str. 6). Prednosti, ki izvirajo iz objektnega pristopa pri izdelavi CIM, so bile inherentne sledenju tega osnovnega namena. IEC 61968 je namenjen integraciji aplikacij pri oskrbi z električno energijo in podpori izdelavi sistemskih vmesnikov, ki naj omogočijo povezljivost aplikacij, ki so ožji del centra vodenja DEES, s tistimi, ki tečejo na poslovnem omrežju nekega podjetja. Tu gre za povezovanje aplikacij, ki so del procesnega omrežja centra vodenja, in aplikacij, ki so del širšega konteksta poslovanja podjetja.

Pri vseh treh standardih gre za določeno mero prekrivanja rešitev in izvedb, tako da danes že najdemo primere poskusov s konkretnimi pilotnimi realizacijami za njihovo integracijo oziroma formalno poenotenje (Kostić, 2003, str. 1-6) ter za oblikovanje scenarijev za izmenjavo podatkov med aplikacijami (Kostić, 2004, str. 1-5).

Uporaba objektno-orientiranih konceptov pomeni bistveno povečano učinkovitost modeliranja procesa in podatkov, učinkovitejše izvajanje nadgradenj in vzdrževanja sistema ter učinkovitejše vodenje DEES. Sprememba klasičnega v objektno orientirani pogled na primarni sistem v konzervativnem okolju elektroenergetike ne bo enostavna, zato je potrebno problematiko širše, predvsem upravljalskim strukturam v podjetjih, predstaviti na način, kjer bodo jasno vidne prednosti predlaganega pristopa pri uporabi na področju operativnega vodenja DEES, pri izobraževanju in logičnem razumevanju delovanja DEES ter posredno pri zmanjševanju operativnih stroškov in povečanju kvalitete dobave električne energije in storitev.

Na področju elektroenergetike vse bolj ozaveščamo tudi dejstvo, da je informatizacija najbolj vitalnih delov poslovanja nujnost. Odpor oziroma vsaj zadržanost do tega je še danes v nekaterih krogih prisotna prav zaradi dejstva, da pri izvajanju javnega interesa na področju distribucije in dobave električne niso dopustna nikakršna naključja in napake, ki bi jih eventualna informatizacija na podlagi »hišnih«, to je neprofesionalnih rešitev, lahko zakrivila. Temu velja sicer delno pritrditi, saj so se take neprijetnosti v svetu že dogajale. Prav zato je potrebno k takim projektom pristopiti premišljeno in predvsem absolutno strokovno, z upoštevanjem vseh bistvenih okoliščin in partikularnosti. Enako pomembna je danes ugotovitev, da je poslovno preživetje podjetij, tudi takih, ki njihov poslovni proces vključuje izvajanje javnega interesa, povezano s spremembo zastarelih vzorcev razmišljanja ter z

⁶⁰ API – Application Program Interface

⁶¹ CIM – Common Information Model

vzpostavitev novega vrednostnega sistema v organizacijski kulturi podjetij. Spremembe vključujejo temeljito preoblikovanje podjetij, ki odpira možnosti, da v novih pogojih poslovanja le-ta zasedejo ugoden konkurenčni položaj, ki bo dolgoročno zadovoljil kupce - odjemalce električne energije in drugih storitev, zaposlene ter upravljske in lastniške strukture podjetij.

7 LITERATURA IN VIRI

LITERATURA

1. Arlow Jim, Neustadt Ila: UML and the Unified Process. Boston: Addison-Wesley, 2002. 395 str.
3. Avison D. E., Fitzgerald G.: Information Systems Development: Methodologies, Techniques and Tools, 2nd ed. London: McGraw-Hill, 1996. 505 str.
4. Booch Grady, Rumbaugh James, Jacobson Ivar: The Unified Modeling language User Guide. Boston: Addison-Wesley, 1999. 482 str.
5. Colnarič Matjaž: Sistemi v realnem času. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2002. 25 str.
6. Črv Milan: Objektni pristop k prenovi poslovnih procesov in izgradnji informacijskega sistema - metodološki vidiki. Doktorska disertacija. Ljubljana: EF, 2000. 202 str.
7. Damij Talib: Tabular Application Development for Information Systems: an object oriented methodology. New York: Springer-Verlag, 2001. 191 str.
8. Debs Atif: Information Systems for The Electric Utility as A Learning Organization. 10th International Conference on Power System Automation and Control PSAC '97, Bled. Ljubljana: Faculty of Electrical Engineering, 1997, str. 73-80.
9. Dilek Murat: Integrated Design of Electrical Distribution Systems: Phase Balancing and Phase Prediction Case Studies. Blacksburg: Faculty of the Virginia Polytechnic Institute and State University, 2001. 141 str.
10. Dobraš Branka, Gudac Goran, Šimunić Juraj: Modeling the Devices Behaviour for Power Substation Control Using Object-Oriented Framework and UML. 11th DAAAM International Symposium, Opatija. Vienna: DAAAM International, 2000, str. 121-122.
11. Douglass Bruce Powel: Real-Time UML. Second Edition. Boston: Addison-Wesley, 2000. 328 str.
12. Douglass Bruce Powel: ROPES: Rapid Object-Oriented Process for Embedded Systems. Reading, MA: I-Logix Inc., 1999. 40 str.
13. Douglass Bruce Powel: Capturing Requirements for Real-Time and Embedded Systems. Reading, MA: I-Logix Inc., 2003. 20 str.
14. Đurović Z., Šimunić Juraj: Power Substation's Domain Modeling with UML – Essential Class Diagrams. European Control Conference ECC'99, Karlsruhe. Düsseldorf: VDI/VDE, 1999, str. 1-6.
15. Fowler Martin, Scott Kendall: UML Distilled. A Brief Guide to the Standard Object Modeling Language. Second Edition. Boston: Addison-Wesley, 2000. 185 str.
16. Gomaa Hassan: Designing Concurrent, Distributed, and Real-Time Applications with UML. Boston: Addison-Wesley, 2000. 785 str.
17. Grad Janez, Jaklič Jurij: Baze podatkov. Ljubljana: Ekonomska fakulteta Univerze v Ljubljani, 1996. 254 str.
18. Gubina Ferdinand, Ogorelec Anton: Vodenje elektroenergetskega sistema. Ljubljana: Sloko CIGRE, 1997. 336 str.
19. Gudac Goran, Dobraš Branka, Šimunić Juraj: Object-Oriented Approach to Power Substation Information Modeling – A Step Towards Efficient System Control. 10th

- International DAAAM Symposium, Vienna. Vienna: DAAAM International, 1999, str. 171-172.
20. Hammer Michael, Champy James: Preurejanje podjetja. Manifest revolucije v poslovanju. Ljubljana: Gospodarski vestnik, 1995. 223 str.
 21. Handschin Edmund et al.: Object-Oriented Based Methods and Techniques in Control Centres Applied to Open Access Schemes. 10th International Conference on Power System Automation and Control PSAC '97, Bled. Ljubljana: Faculty of Electrical Engineering, 1997, str. 167-172.
 22. Hebel Z. et al.: Control Centre Operation Improvement by Modern Technology. 10th International Conference on Power System Automation and Control PSAC '97, Bled. Ljubljana: Faculty of Electrical Engineering, 1997, str. 45-48.
 23. Hrovatin Nevenka et al.: Strategija diverzifikacije Elektro Primorska d.d. Ljubljana: Eco Consulting, 2003. 179 str.
 24. Jacobson Ivar: Object-Oriented Software Engineering: A Use Case Driven Approach. Workingham: Addison-Wesley, 1996. 528 str.
 25. Klom Andy: Deregulation of electricity in the European Union. Energy in Europe. Luxemburg: Official Publications of the European Communities, 1997, 28. str. 86 - 95.
 26. Kostić Tatjana, Preiss Otto, Frei Christian: Towards the Formal Integration of Two Upcoming Standards: IEC 61970 and IEC 61850. LESCOPE Conference, Montreal. Paris: CIGRE, 2003, str. 1-6.
 27. Kostić Tatjana et al.: Scenarios for Data Exchange using Standards IEC 61970 and IEC 61850. UCA User Group meeting, Paris. Paris: CIGRE, 2004, str. 1-5.
 28. Kovačič Andrej: Informatizacija poslovanja. Ljubljana: EF, 1998. 214 str.
 29. Kovačič Andrej, Vintar Mirko: Načrtovanje in gradnja informacijskih sistemov. Ljubljana: DZS, 1994. 316 str.
 30. Lipovec Rado: Inženiring zahtev z uporabo enotnega modelirnega jezika UML in metode TAD. Magistrsko delo. Ljubljana: Ekonomska fakulteta Univerze v Ljubljani, 1998. 92 str.
 31. Martin James: Principles of Object-Oriented Analysis and Design. Englewood Cliffs: Prentice Hall, 1993. 412 str.
 32. Midttun Atle: European energy industry business strategies. Amsterdam: Elsevier, 2001. 427 str.
 33. Minio E.E.: Methodology for the Development and Implant of Information Systems into an Electric Energy Distribution Company. 16th International Conference on Electricity Distribution, Amsterdam. London: CIRED Institution of Electrical Engineers, 2001, str. 1-5.
 34. Možina Stane et al.: Management. Radovljica: Didakta, 1994. 1072 str.
 35. Ottosson H. et al.: The ISES project book. Malmö: Enersearch AB Consortium, 1998. 221 str.
 36. Podmore Robin et al.: Common Information Model – A Developer's Perspective. Proceedings of the 32nd Hawaii International Conference on System Sciences. New York: IEEE, 1999, str. 1-6.
 37. Quatrani Terry: Introduction to the Unified Modeling Language. Rational Developer Network. New York: IBM Rational developer community, 2001, 20 str.
 38. Randić Mirko, Šimunić Juraj, Morović Mira: Information Modeling of Power Substations by Using the Unified Modeling Language. 10th Mediterranean Electrotechnical Conference, Lemesos. Nicosia: IEEE Cyprus section, 2000, str. 995-998.
 39. Ratej Jure: Odpiranje trga z energijo. Ljubljana: Korona d.d., 2000. 16 str.

40. Rumbaugh James, Jacobson Ivar, Booch Grady: The Unified Modeling Language Reference Manual. Reading: Addison-Wesley, 1999. 550 str.
41. Soman S.A., Khaparde S.A., Pandit Shubha: Computational methods for large sparse power systems analysis: An object oriented approach. Boston: Kluwer Academic Publishers, 2002. 360 str.
42. Šimunić Juraj, Đurović Z., Gudac Goran: Object-Oriented Approach to Analysis in Power Substation Control. 10th International Conference on Power System Automation and Control PSAC '97, Bled. Ljubljana: Faculty of Electrical Engineering, 1997, str. 203-207.
43. Šimunić Juraj, Randić Mirko, Živić Marijana: A Device-Based Process Signal Design of the Electric Power Plants. Journal of Computing and Information Technology. Zagreb: University Computing Centre, 2001, str. 29-42.
44. Šimunić Juraj, Randić Mirko, Živić Marijana: Object-Oriented Information Modeling of Power Substations. 9th International DAAAM Symposium, Cluj-Napoca. Vienna: DAAAM International, 1998, str. 441-442.
45. Watson Gregory H.: Business Systems Engineering. New York: John Wiley & Sons, 1994. 287 str.

VIRI

1. Annual Report 1999. Ljubljana: ELES, 1999. 59 str.
2. Baze podatkov – študijsko gradivo. Ekonomska fakulteta Ljubljana (doc.dr. Jurij Jaklič), 1999. 164 str.
3. DIN 44300. Berlin: Deutsches Institut für Normung. Informationsverarbeitung-Begriffe, 1988.
4. Directive 96/92/EC of the European Parliament and of the Council (Official journal of the European Communities, No. L 027).
5. Directive 2003/54/EC of the European Parliament and of the Council (Official journal of the European Communities, No. L 176/37).
6. Energetski zakon (Uradni list RS, št. 79/99).
7. Idejni projekt DCV Elektro Primorska. Ljubljana: Elektro Primorska - Ensico, 1999. 197 str.
8. Letno poročilo družbe Elektro Primorska d.d. za leto 2002. Nova Gorica: Elektro Primorska, 2003. 77 str.
9. OMG Unified Modeling Language Specification. [URL: <http://www.omg.org/docs/formal/03-03-01.pdf>], Object Management Group, 10.01.2005.
10. Osnovni OO koncepti. [URL: <http://lisa.uni-mb.si/cot/tutoriali/omt/koncepti.htm>], FERi Univerza v Mariboru, 11.01.2005.
11. Pomen jezika za objektno modeliranje UML. [URL: <http://lisa.uni-mb.si/cot/cotl/98jesen/uml.htm>], FERi Univerza v Mariboru, 11.01.2005.
12. Predstavitev organiziranosti Elektro Primorska d.d. [URL: <http://www.elektro-primorska.si/predstavitev/slo/organiziranost.htm>], Elektro Primorska, 07.11.2003.
13. Projekt integriranega modela podatkov. Maribor: Elektrogospodarstvo Slovenije, 1989. 251 str.

14. SIST EN 61850:2004. Communication networks and systems in substations. Ljubljana: Slovenski inštitut za standardizacijo, 2004.
15. SIST EN 61968:2004. Integracija aplikacij pri oskrbi z električno energijo – Sistemski vmesniki za upravljanje omrežja. Ljubljana: Slovenski inštitut za standardizacijo, 2004.
16. SIST EN 61970:2004. Energy management system application program interface (EMS-API) – Common Information Model (CIM). Ljubljana: Slovenski inštitut za standardizacijo, 2004.
17. Statistični letopis energetskega gospodarstva RS 2001. Ljubljana: Ministrstvo za okolje, prostor in energijo, 2002. 200 str.
18. Upravljanje distribucijskega omrežja. [URL: <http://www.elektro-primorska.si/upravljanje/slo/upravljanje.htm>], Elektro Primorska, 21.07.2003.
19. Uredba o načinu izvajanja javnih gospodarskih služb s področja distribucije električne energije (Uradni list RS št. 54/00, 31/01, 99/01).

PRILOGA

- A. Funkcionalna dekompozicija delovnih procesov
- B. Začetni diagram razredov
- C. Končni diagram razredov
- D. Diagrama stanj
- E. Zaporednostna diagrama
- F. Diagrama aktivnosti
- G. Seznam uporabljenih kratic

A. Funkcionalna dekompozicija delovnih procesov

Vodenje obratovalnega stanja DEES

AKTIVNOSTI	POSILI	OBSTOJEČA INFORMACIJSKA PODPORA
Izdajanje komand za spremembo vklopnih stanj	- Izbira elementov krmiljenja. - Preverjanje kompetenc krmiljenja. - Preverjanje stanja elementov krmiljenja. - Preverjanje zapahovanj (blokad) krmiljenja. - Izdaja in potrjevanje izdaje komand elementom krmiljenja. - Preverjanje in potrjevanje izvedbe spremembe vklopnega stanja.	DA
Vodenje sprememb vklopnih stanj	- Avtomatsko javljanje in registracija vseh sprememb vklopnih stanj elementov DEES. - Ročni vnos in registracija vseh sprememb vklopnih stanj elementov DEES.	DA
Nastavljanje zelenih vrednosti parametrov	- Nastavljanje zelenih vrednosti napetosti. - Nastavljanje načina regulacije napetosti (ročno, avtomatsko). - Krmiljenje napetosti. - Krmiljenje porabe z MTK komandami. - Nastavljanje zelenih vrednosti delovanja zaščit.	delno DA (zahtevana dopolnitev)
Ocenjevanje in omejevanje porabe in obremenitev DEES	- Izvajanje omejevanja porabe električne energije z nastavljanjem zelene vrednosti napetosti. - Izvajanje omejevanja porabe električne energije na podlagi planov omejevanja (izklopi odjemalcev).	NE (zahtevana)
Vodenje obratovalno - tehnične dokumentacije	- Vodenje obratovalnih in tehničnih navodil. - Vodenje dnevnikov poročil, dogodkov, motenj in nenormalnih stanj DEES. - Vodenje knjig depeš in navodil za varno delo. - Vodenje knjig prijav in odobritev planiranih del. - Vodenje knjig obiskov, predaje ključev ipd.	NE (zahtevana)

Planiranje porabe in obremenitev DEES

AKTIVNOSTI	POSILI	OBSTOJEČA INFORMACIJSKA PODPORA
Napovedovanje in planiranje obremenitev	• Izbira tipa napovedi-voznega reda (dnevni, tedenski, mesečni, letni). • Izbira tipa odjema. • Uporaba historičnih podatkov odjema. • Uporaba nadomestnih obremenilnih diagramov. • Izbira vplivnih parametrov (vreme, sezona, karakteristični dan). • Zbiranje in sestavljanje obremenilnih diagramov glede na bilančne skupine, območje, RTP itd. • Izračun oziroma priprava napovedi voznega reda za odjemalca, izvod omrežja, RTP, bilančno skupino, preskrbovalno področje.	NE (zahtevana)
Primerjava med napovedano in dejansko porabo ter izračun odstopanj	• Zajem dejanskih podatkov obremenitev znotraj izbranega časovnega okna za tip napovedanega voznega reda. • Periodični zajem in tekoče spremljanje vrednosti obremenitev in sprotne primerjava z napovedjo - generiranje alarmov ob prekoračitvi pragov odstopanj. • Obdelava (primerjava) napovedanih obremenitev z dejanskimi za potrebe komercialnih izračunov odstopanj in penalov za prekoračitve. • Priprava poročil o odstopanjih.	NE (zahtevana)
Planiranje omejevanja porabe	• Izvajanje nadzora nad parametri obratovanja DEES. • Določitev kriterijev za izvajanje omejevanja porabe. • Definiranje prioritete in načine krmiljenja napetosti. • Definiranje parametrov delovanja podfrekvenčne zaščite in SN izvodov vključenih v to zaščito. • Definiranje prioriteta izklopov odjemalcev. • Definiranje postopkov ob izpadih večjih agregatov v sistemu.	NE (zahtevana)

Izvajanje analiz obratovalnih stanj in dogodkov ter optimalnega obratovanja DEES

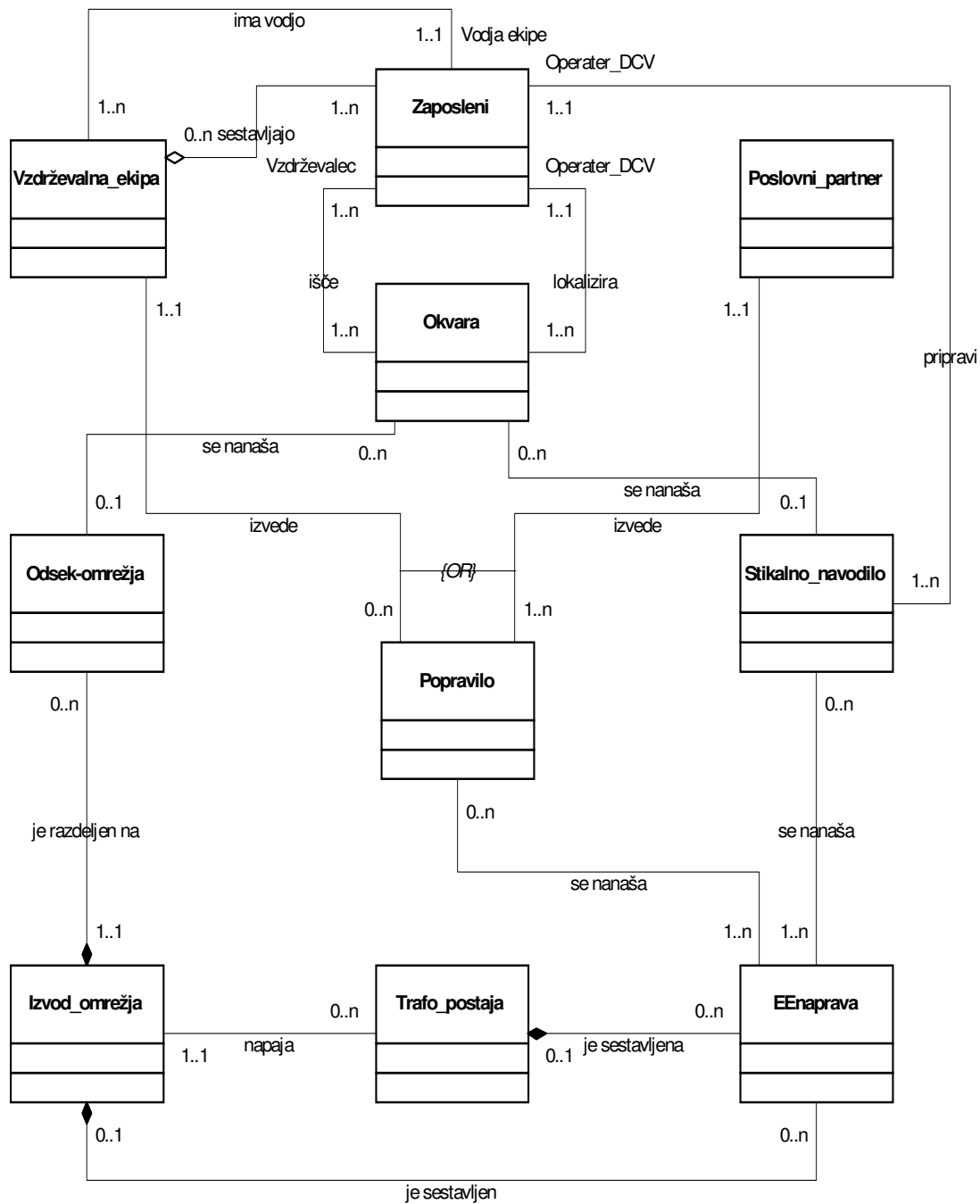
AKTIVNOSTI	POSILI	OBSTOJEČA INFORMACIJSKA PODPORA
Analiza obratovalnih stanj in dogodkov	- Analiza obremenitev in napetostnih razmer (simulacije, uporaba trenutnih podatkov, simuliranje vklopnih stanj in dogodkov, izračuni). - Analiza dogodkov in stanj v preteklem obdobju (simulacije na podlagi dejanskih podatkov iz preteklega obdobja). - Analiza dogodkov in izdelava statistike dogodkov na DEES po raznih parametrih in za različne potrebe in uporabnike.	delno DA (zahtevana dopolnitev)
Izračun optimalnega vklopnega stanja DEES	- Izbira dela omrežja DEES za izračun. - Izbira tipa - kriterija izračuna optimalnega vklopnega stanja (min. izgub, upoštevanje max.obremenitev, upoštevanje napetostnega profila v omrežju). - Vnos robnih pogojev in omejitev izračuna. - Izračun optimalnih pretokov energije, izgub, napetosti. - Določitev razklopnih mest v omrežju.	delno DA (zahtevana dopolnitev)
Izračun kratkih stikov v DEES	- Izbira mesta okvare in tipa okvare. - Določitev robnih pogojev izračuna. - Izračun kratkostičnih tokov in kratkostične moči v DEES. - Prikaz kritičnih vrednosti glede na vgrajeno opremo in zaščitne naprave.	delno DA (zahtevana dopolnitev)
Izračun nastavitve regulacije napetosti DEES	- Izbira lokalne ali globalne nastavitve napetostnih nivojev. - Vnos parametrov nastavitve in regulacije napetosti. - Vnos pragov dovoljene napetosti na delih DEES in napravah. - Izračun nastavitve regulacijskih stopenj transformatorjev glede na vnesene parametre in kriterije. - Prikaz kritičnih vrednosti in preseženih spodnjih in zgornjih pragov napetosti v DEES, pri odjemalcih in na napravah.	NE (zahtevana)
Izračun padcev napetosti v DEES	- Izbira dela DEES, za katerega se izvaja izračun. - Vnos zgornjih mej padcev napetosti. - Izbira "on-line" oziroma historičnih podatkov o obremenitvah za potrebe izračuna padcev napetosti. - Izračun padcev napetosti. - Prikaz odstopanj napetosti na kritičnih mestih in opsijsko izračun nastavitve regulacijskih odcepov transformatorjev.	delno DA (zahtevana dopolnitev)

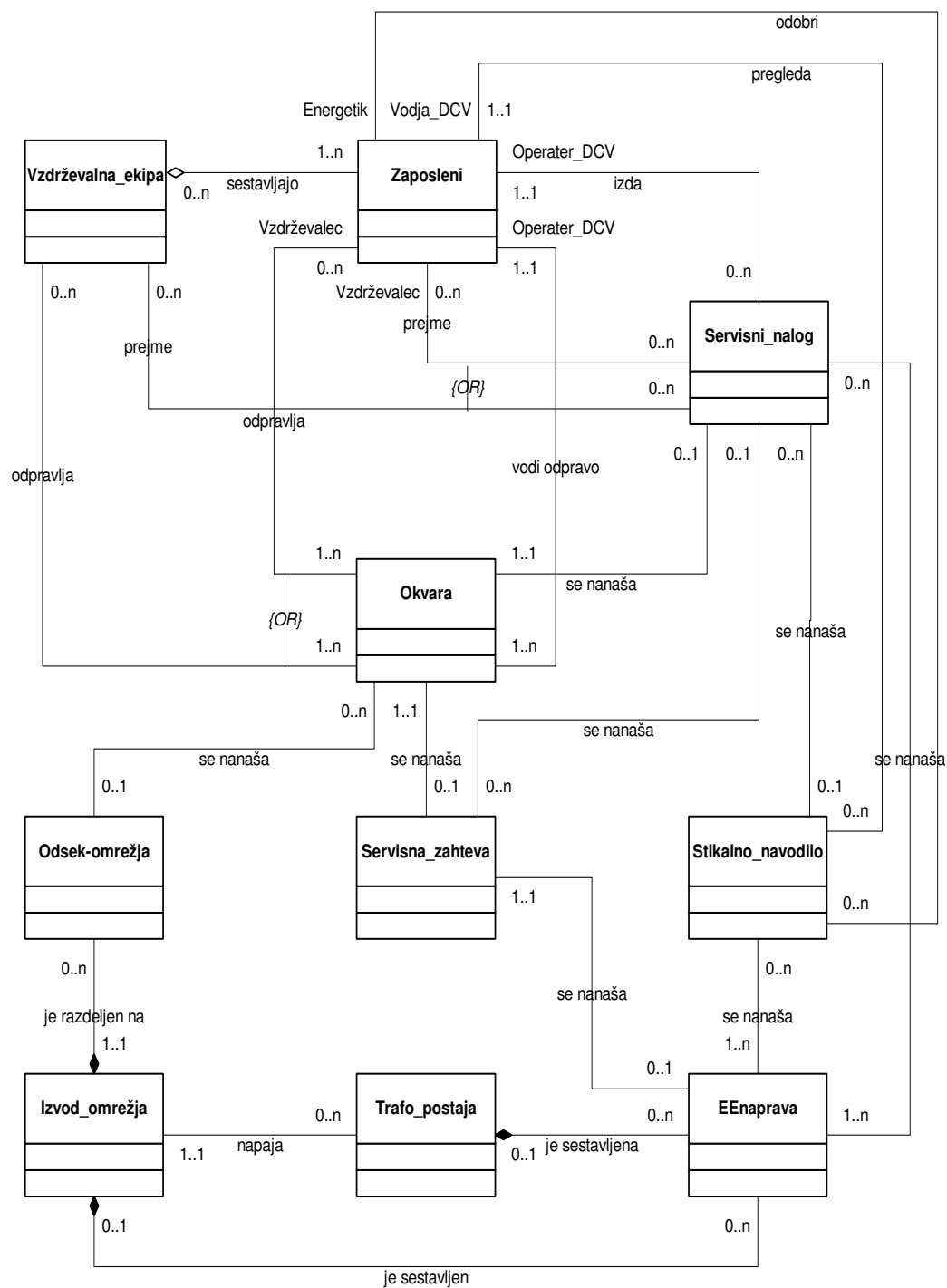
Izračun vključevanja transformatorjev	- Izbira dela DEES oziroma RTP, kjer se izvaja izračun optimalnega obratovanja transformatorjev. - Izbira obremenilnega diagrama TR za neko določeno obdobje obratovanja. - Določitev kriterijev za izbiro optimalnega vklopnega stanja TR (min. izgub v TR, min. stroškov obratovanja, dovoljena preobremenitev TR, dovoljena kratkostična moč zbiralnic, zanesljivost).	NE (zahtevana)
--	---	--------------------------

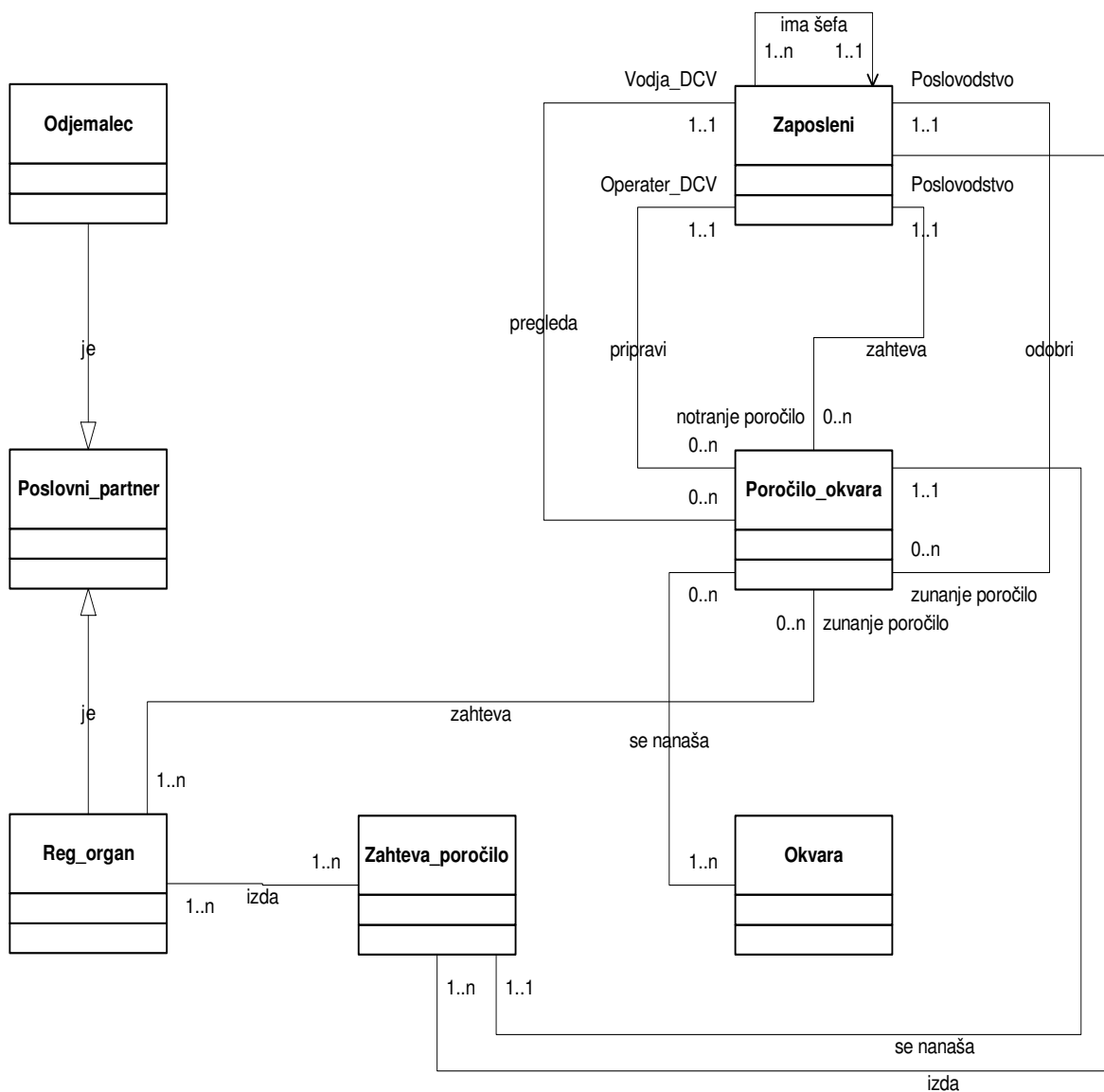
Izvajanje obdelave okvar v DEES

AKTIVNOSTI	POSILI	OBSTOJEČA INFORMACIJSKA PODPORA
Prijava okvare	- Prijava okvare. - Evidentiranje okvare in odjemalca. - Korelacija okvare. - Generiranje servisne zahteve. - Obveščanje odjemalca.	NE (zahtevana)
Lokalizacija okvare	- Določitev področja okvare. - Priprava, pregled in posredovanje stikalnega navodila zadolženim za odpravo okvare. - Izvajanje stikalnega navodila za lokalizacijo okvare. - Zaključek del po navodilu in potrditev zaključka del in lokalizacije okvare.	NE (zahtevana)
Odprava okvare	- Priprava servisnega naloga in stikalnega navodila. - Pregled in odobritev stikalnega navodila. - Posredovanje servisnega naloga in stikalnega navodila zadolženim za odpravo okvare. - Izvajanje servisnega naloga za odpravo okvare. - Zaključek del po servisnem nalogu in potrditev zaključka del in odprave okvare. - Obveščanje odjemalca o odpravi okvare, ki je vezana nanj oziroma katere prijava je bila sprožena s strani odjemalca.	NE (zahtevana)
Obveščanje in poročila	- Izdaja zahteve za poročilo s strani različnih uporabnikov. - Evidentiranje zahteve s strani operaterja DCV. - Priprava in odobritev poročila. - Odprava poročila do uporabnikov oziroma zahtevnikov.	NE (zahtevana)

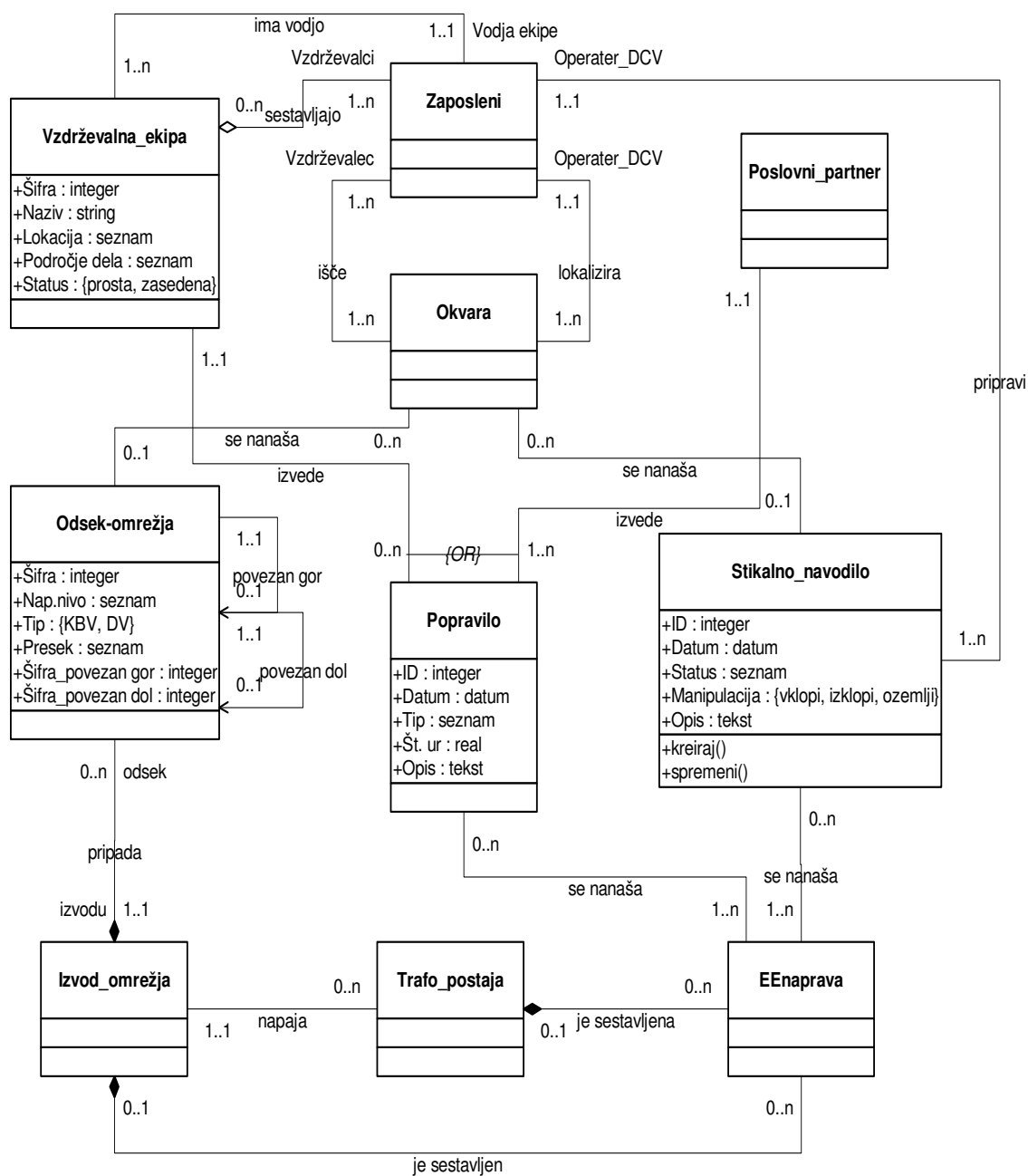
B. Začetni diagram razredov

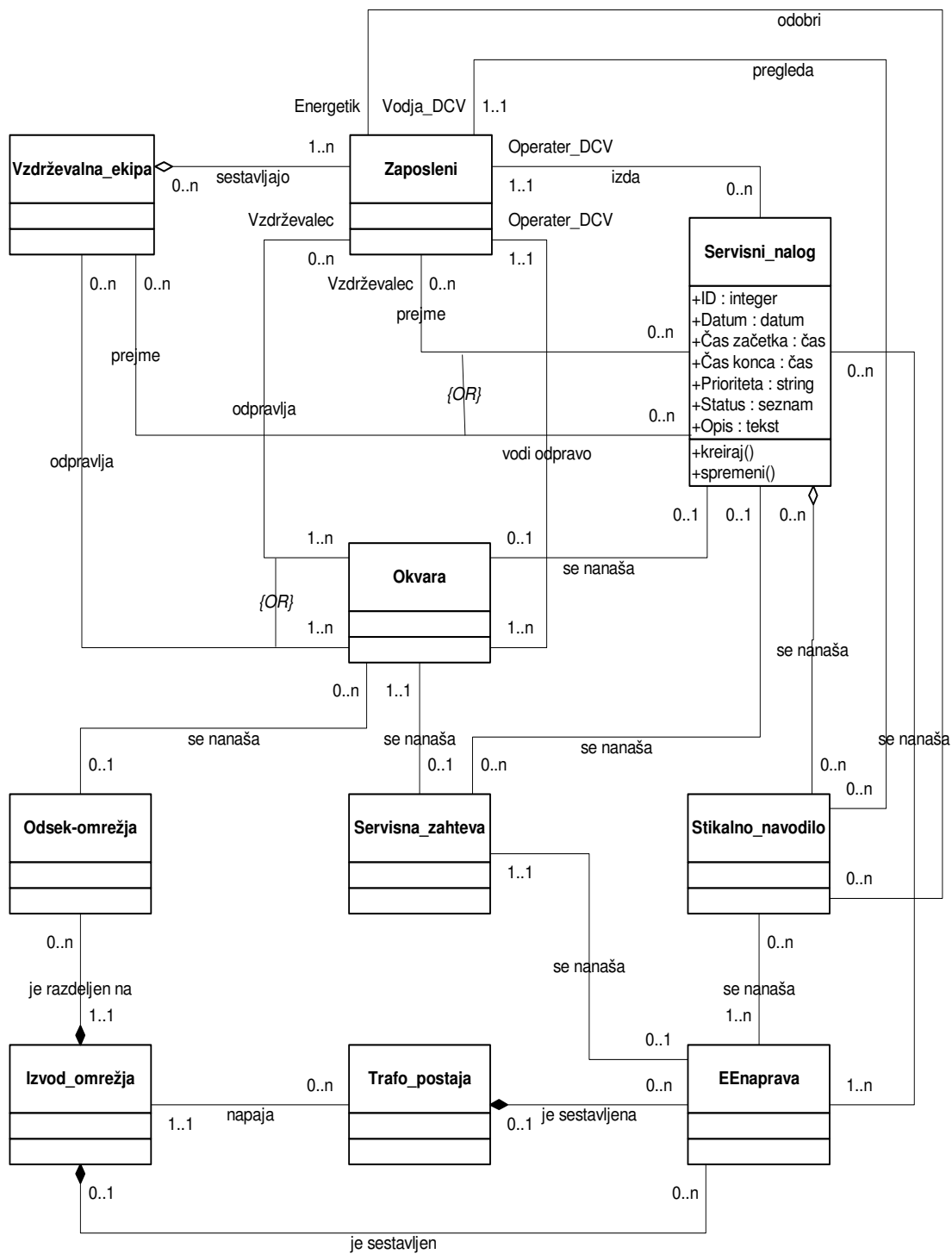




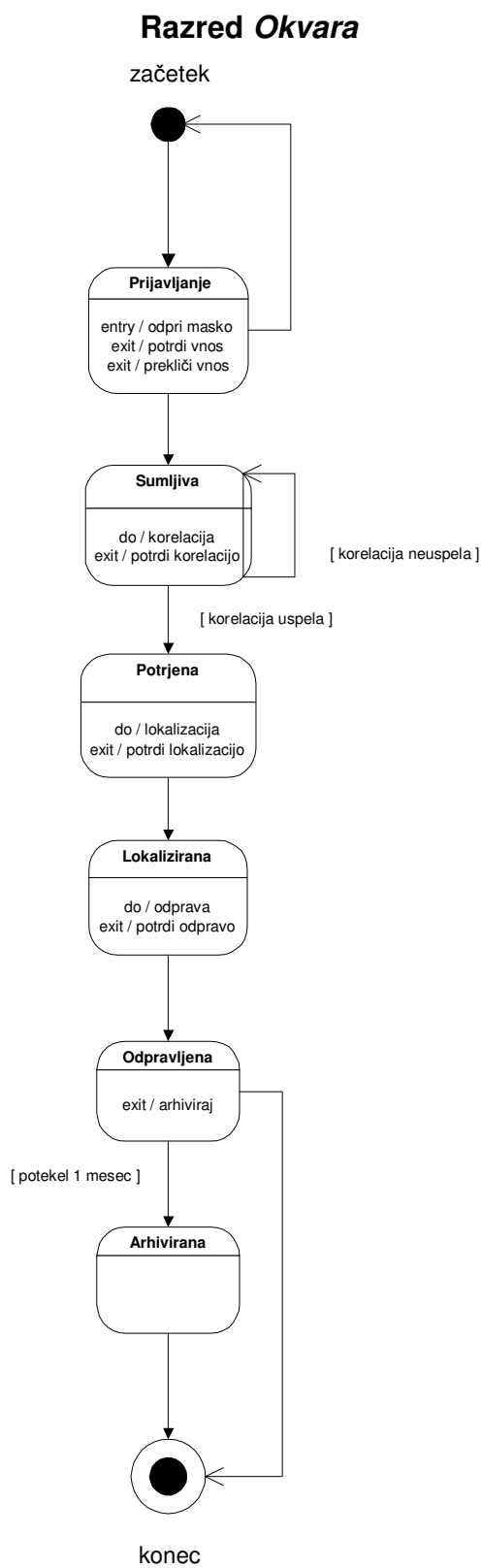


C. Končni diagram razredov

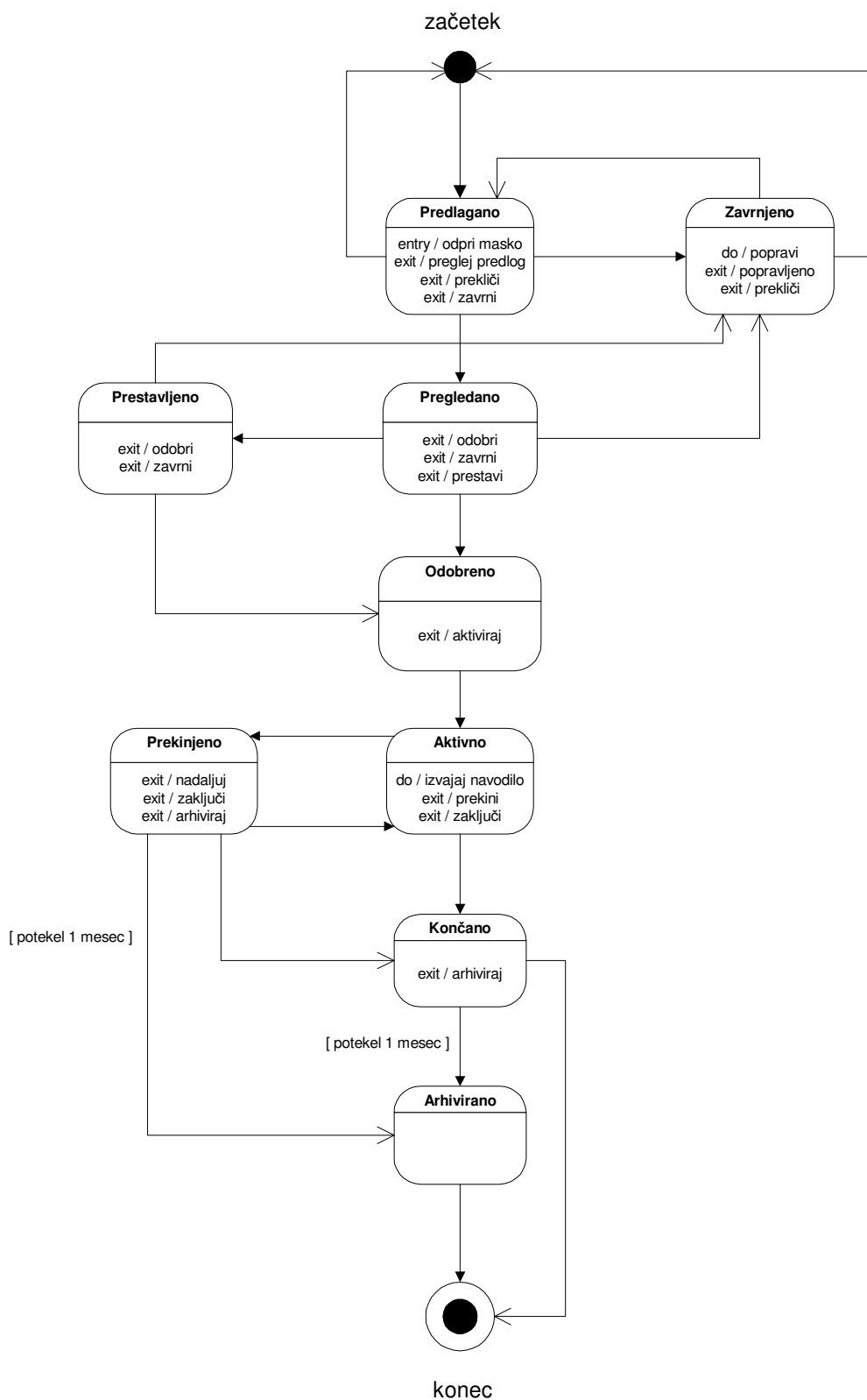




D. Diagrama stanj

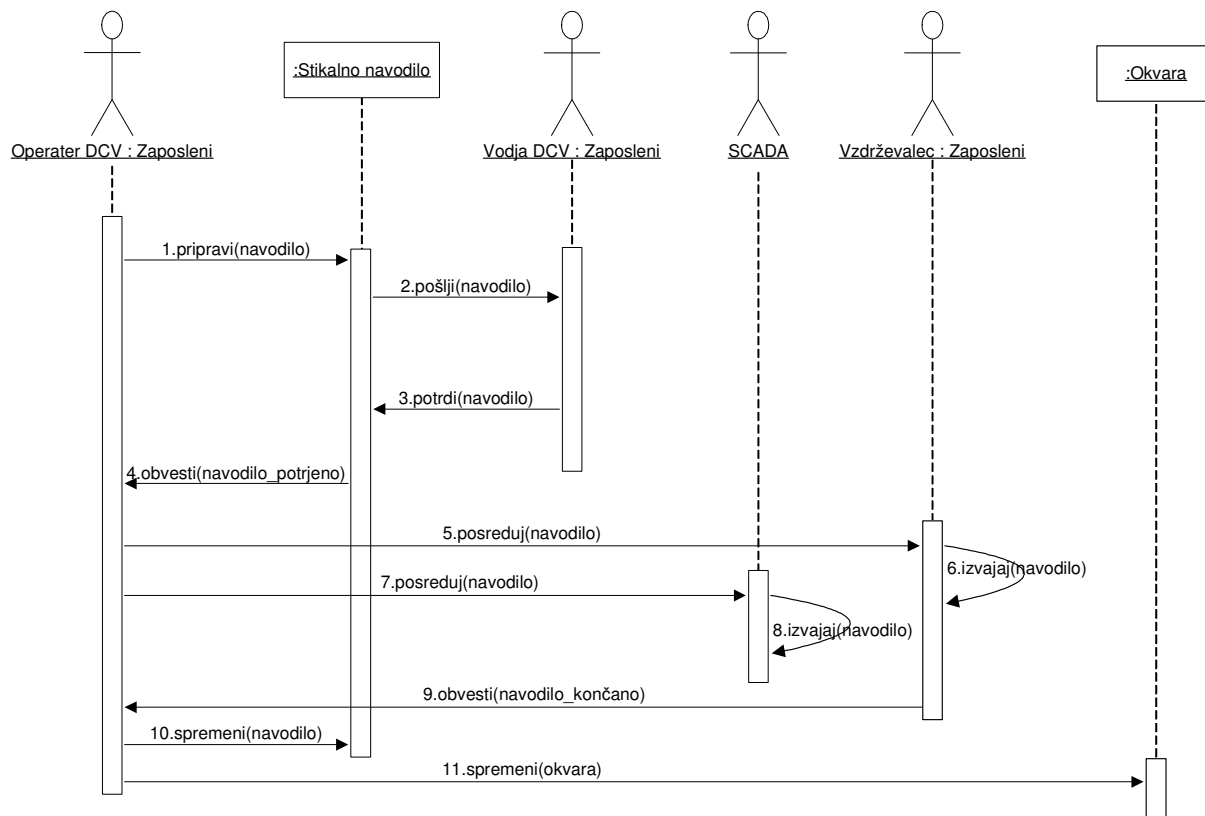


Razred *Stikalno navodilo*

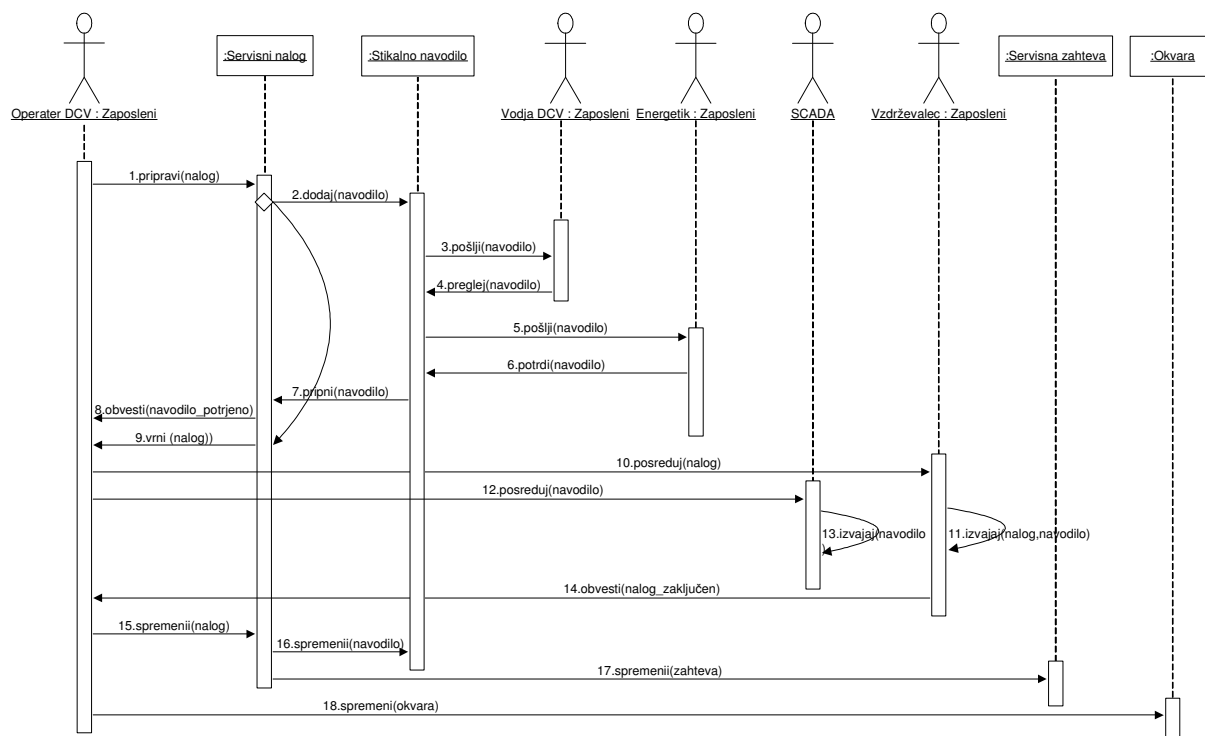


E. Zaporednostna diagrama

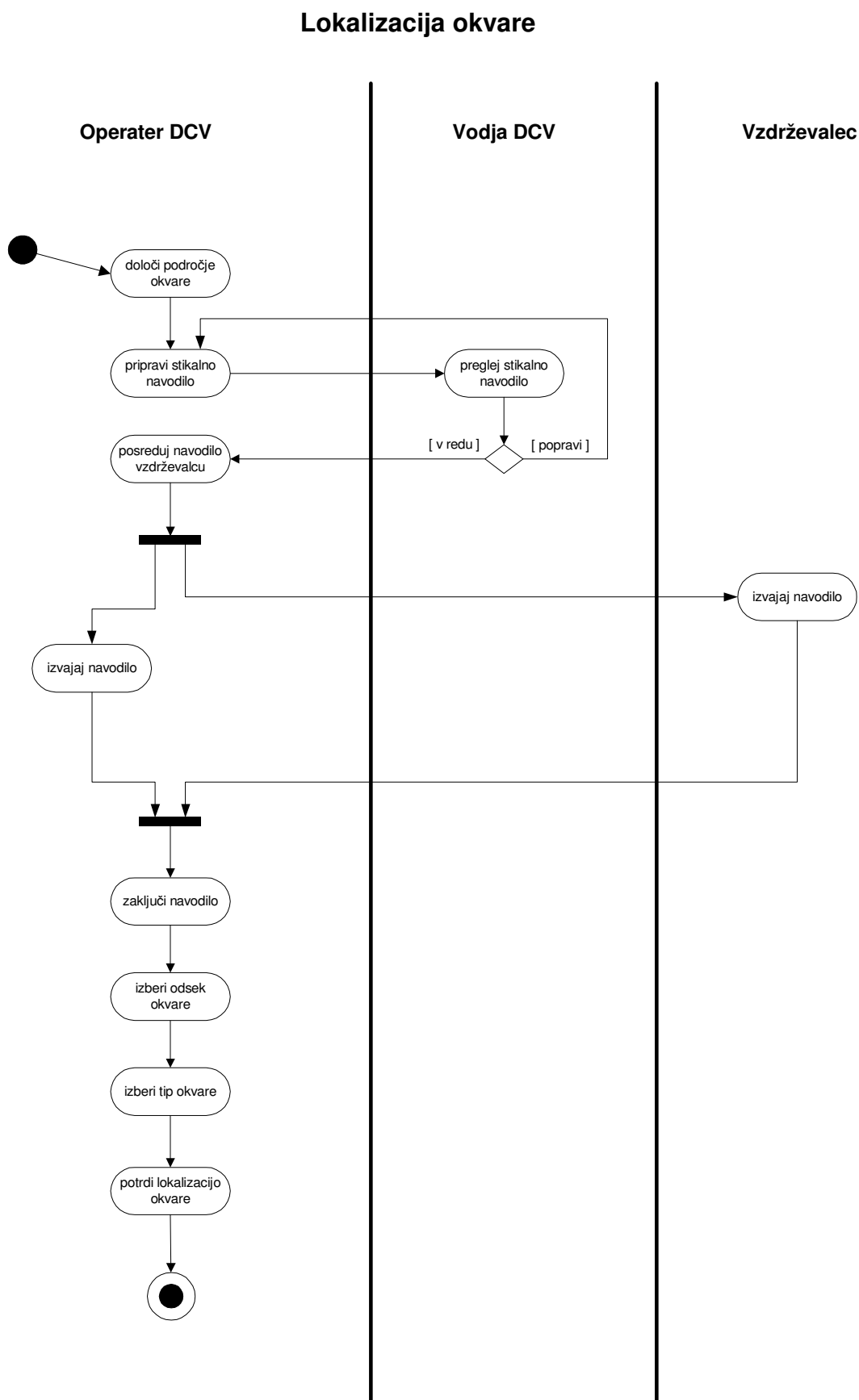
Lokalizacija okvare



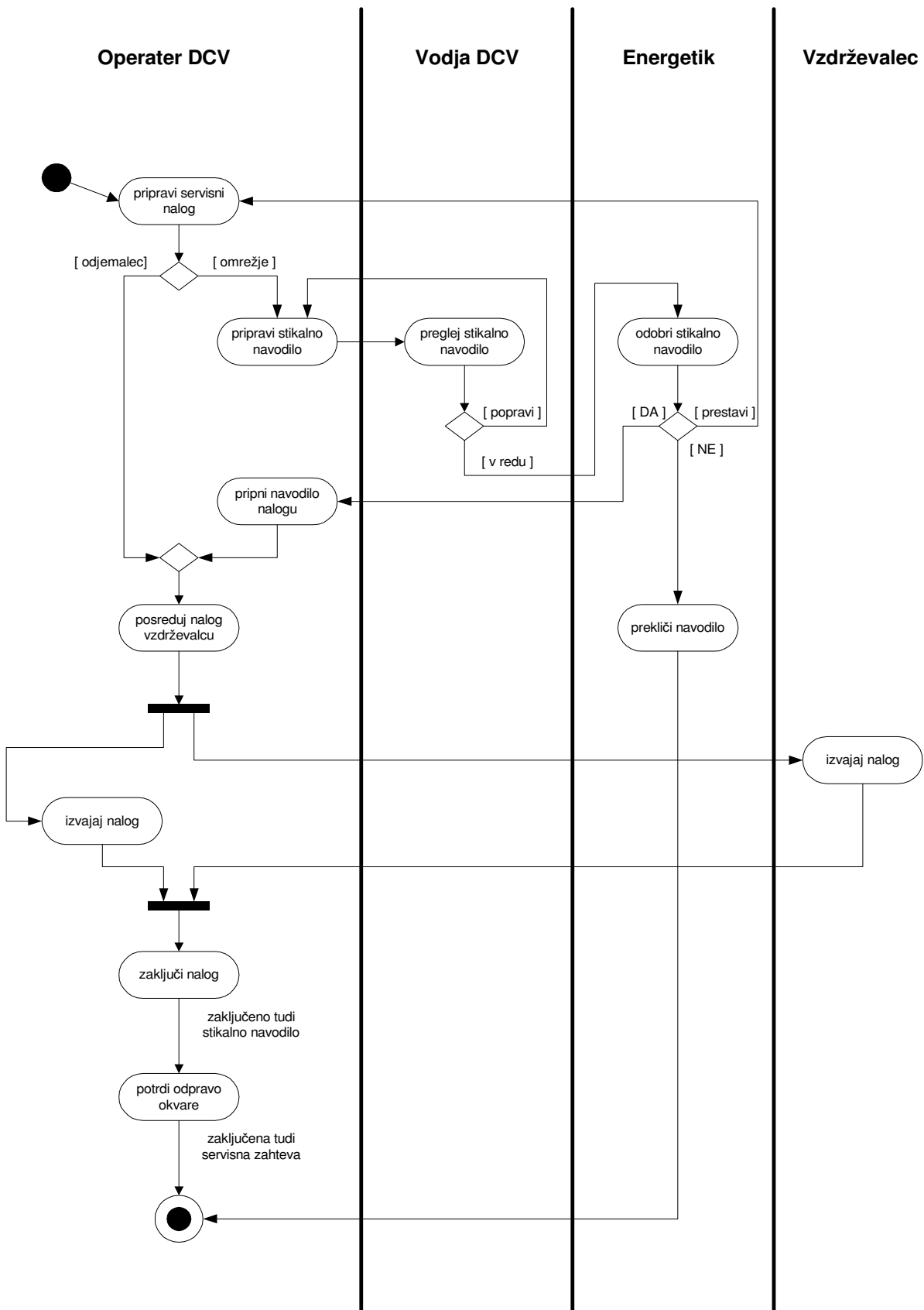
Odprava okvare



F. Diagrama aktivnosti



Odprava okvare



G. Seznam uporabljenih kratic

API - Application Program Interface
CIM - Common Information Model
CV - center vodenja
DCV - Distribucijski center vodenja
DEES - distribucijski elektroenergetski sistem
DMS - Distribution Management System
DV - daljnovod
DVP - daljnovodno polje
EE - elektroenergetski
EEP - elektroenergetski postroj
EES - elektroenergetski sistem
ELES - Elektro Slovenija, d.o.o.
EP - Elektro Primorska, d.d.
EZ - Energetski zakon
GJS - gospodarska javna služba
GWh - (giga)vatna ura
HMI - Human Machine Interface
IEC - International Electrotechnical Commission
IL - izvodni ločilnik
ISO - International Organization for Standardization
IT - Information Technology
KBV - kablovod
kHz - (kilo)hertz
KO - krmilna omarica
KRO - krmilno-relejna omara
KZP - krmilno-zaščitni panel
kV - (kilo)volt
kWh - (kilo)vatna ura
MTK - mrežno tonsko krmiljenje
MWh - (mega)vatna ura
NN - nizkonapetostni
NPC - nadzorni računalnik
OCL - Object Constraint Language
OKC - operativno-komunikacijski center
OMG - Object Management Group
OMT - Object Modeling Technique
OO - objektno-orientirani
OOSE - Object Oriented Software Engineering
PL - panel napajanja lastne rabe
PP - požarni panel
ROA - Return on Assets
ROE - Return on Capital
RTP - razdelilna transformatorska postaja

RTS - Real Time System
RTU - Remote Terminal Unit
SCADA - Supervisory, Control and Data Acquisition
SDV - sistem daljinskega vodenja
SN - srednjenapetostni
SO - sinoptična plošča
TK - telekomunikacije (telekomunikacijski)
TQM - Total Quality Management
TR - transformator
UDO - upravljalec distribucijskega omrežja
UPO - upravljalec prenosnega omrežja
UML - Unified Modeling Language
VN - visokonapetostni