

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**ZASNOVA INFORMACIJSKEGA SISTEMA ZA UPRAVLJANJE Z RAZVOJNIMI
PODATKI IN DOKUMENTACIJO IZDELKOV**

V Ljubljani, september 2004

IZJAVA

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

V Ljubljani, dne 30.09.2004

Podpis:

KAZALO

1. Uvod	1
2. PDM-sistemi.....	5
2.1 Mesto in vloga PDM-sistema v podjetju.....	5
2.2 Povezovanje med PDM in ERP sistemi	11
2.3 Zagotavljanje kakovosti (ISO) in informacijski sistem	15
2.4 Osnovna funkcionalnost PDM sistemov	17
3. UML	26
3.1 ZGODOVINA	26
3.2 NAMEN MODELIRANJA Z JEZIKOM UML	27
3.3 KONCEPTUALNI MODEL	28
4. Intervju.....	42
5. Načrtovanje PDM sistema v podjetju s pomočjo UML metode.....	43
5.1 Primeri uporabe.....	43
5.2 Interakcijski diagrami.....	45
5.3 Razredni diagrami.....	49
5.4 Aktivnostni diagrami.....	51
6. Uvajanje in nastavitev PDM sistema s pomočjo UML.....	53
6.1 Definicija delovnih map v PDM sistemu	53
6.2 Definicija projektnih map in njene strukture.....	57
6.3 Definicija dokumentov in vrst dokumentov	59
6.4 Elektronsko usmerjanje dokumentacije in obveščanje.....	61
6.5 Vodenje sprememb in sledljivost	63
7. Povezovanje med PDM in ERP sistemom Baan v podjetju	63
7.1 Projektno vodenje v Baan ERP sistemu	64
7.2 Delovanje vmesnika med PDM sistemom in ERP sistemom v razvojni fazi projekta.....	68
7.3 Zaključevanje projekta uvajanja informacijskega sistema.....	69
8. Povezovanje med PDM sistemom in MS Office aplikacijami.....	70
9. Sklep	72
10. Literatura	74
11. Viri.....	76
12. Priloga.....	78

1. Uvod

Globalizacija in tržišče narekujeta vedno krajši razvojni cikel in boljše kvaliteto produktov z vse bolj zahtevno strukturo. Ob tem narekujeta vedno nižje cene izdelkov in hkrati visoko stopnjo prilagodljivosti posameznim kupcem. Ključno vlogo pri razvoju imata računalniška in informacijska tehnologija, ki nenehno ponujata nove možnosti. Količine podatkov in dokumentov, ki so na voljo so vedno večje in težko obvladljive na klasičen način vodenja dokumentacije. Zaradi kompleksnosti potrebujemo sistematičen pristop, spremembe v organizaciji in različna znanja od informatike, metodike dela v razvoju, komunikacije in dela z ljudmi [Duhovnik, 2000, str. 1.1]. Prvi korak pri prenovi informacijskega sistema predstavlja analiza informacijskih tokov in poznavanje poteka dela [Belbin, 1993, str. 86]. Poslovni in proizvodni informacijski sistemi se uspešno uporabljajo že vrsto let in so v podjetjih dobro podprti. Na voljo je vse več orodij in metod za hitrejše uvajanje teh sistemov [Scheer, 1994, str. 106], [Bernus, 1998, str. 345]. V zadnjem času je tehnologija dozorela za informacijsko povezovanje znotraj tehničnih oddelkov v podjetjih [Bernus, 1996, str. 244].

Razvoj novih izdelkov ustvarja veliko stalno spreminjajočih se podatkov in zato zahteva specifičen pristop. V podjetjih, kjer je potrebno obvladovati veliko količino podatkov in koordinirati delo ljudi, lahko uporabimo ustrezen sistem za upravljanje s podatki in delom v razvoju (PDM sistem – Product Data Management). V večjih sistemih je reorganizacija neizogibna in prednosti, ki jih prinese uporaba PDM sistema hitro opravičijo stroške investicije [Duhovnik, 2000, str. 1.1]. Elektronska oblika dokumentov omogoča postavitve sistema z bistveno boljšo odzivnostjo [Miller, 1998, str. 2]. Z razvojem tehnologije in padanjem cen opreme postajajo takšni sistemi (PDM sistemi oz. PLM sistemi – Product Lifecycle Management) zanimivi tudi za majhna in srednja podjetja [Corbett, 1991, str. 178]. Uvajanje takšnega sistema je lahko uspešno le ob sistematičnem poteku in ob upoštevanju celovitosti problema [Flood, 1994, str. 46]. Podjetje mora najprej razpoznati, kaj je v zvezi s pretokom informacij bistveno za učinkovitost podjetja [Duhovnik, 2000, str. 1.1].

Z uporabo računalniško podprtih tehnologij je prišlo do povečanja količine informacij. Ustrezno vpeljan računalniški sistem za podporo razvoju (PDM oz. PLM) predstavlja rešitev v novonastalem položaju [Duhovnik, 2000, str. 1.2]. PDM sistem podpira upravljanje s celotnimi podatki o izdelku oziroma vso ostalo dokumentacijo (vse vrste dokumentov na nivoju podjetja – projektna dokumentacija...) in aktivnostmi na nivoju celotnega podjetja in hkrati predstavlja osnovo za kontroliran tok podatkov skozi celotno življenjsko dobo izdelka. S

pomočjo PDM sistema lahko zagotovimo spremljanje nastajanja dokumentacije v skladu s standardi o zagotavljanju kakovosti (ISO...) in omogočiti dostop do prave verzije dokumentov vsem ostalim službam (nabava, prodaja...) [Duhovnik, 2000, str. 1.2].

PDM sistem predstavlja metodologijo in potrebno programsko opremo, ki omogoča vodenje podatkov o proizvodih in podpira razvojni proces. Zajema pregledno hranjenje vseh podatkov, ki definirajo izdelek skozi vse faze življenjskega cikla. Poleg mehanizmov iskanja, klasifikacije in arhiviranja omogoča tak sistem tudi kontroliran potek dokumentov in obveščanje med uporabniki (workflow) [Duhovnik, 2000, str. 1.3]. Prav tako pomembna je tudi možnost sledljivosti sprememb s pomočjo verzij dokumentov, statusov teh verzij (veljavna je lahko samo ena verzija) in opomb z razlogi kreiranja nove verzije.

PREDNOSTI UPORABE PDM SISTEMA SO [Prodan, 2004, str.389]:

Prednosti za podjetje:

- Centralno arhiviranje dokumentacije izboljša pristop in komunikacijo v podjetju ter odpiranje novih možnosti sodelovanja s poslovnimi partnerji;
- Znanje podjetja je dostopno širokemu krogu ljudi;
- PDM - sistem je okolje, ki ima vgrajena pravila zagotavljanja kakovosti in s tem omogoča praktično izvajanje zahtev standardov (ISO 9000/2000);
- Hiter dostop do potrebnih podatkov omogoča sprejemanje pravih odločitev;
- Elektronska oblika podatkov omogoča sočasen način dela;

Prednosti za uporabnika:

- Enoten uporabniški vmesnik za različno programsko opremo;
- Bogata možnost iskanja podatkov in dokumentov ter njihova preglednost;
- Zmanjšanje rutinskega dela in bolj učinkovito komuniciranje uporabnikov;

Prednosti za proizvode:

- Pregledno hranjenje podatkov in dokumentov proizvoda;
- Spremljanje vseh sprememb skozi vse faze življenjskega cikla proizvoda;
- Standardizacija proizvodov in njihova ponovna uporaba. S tem dosežemo izboljšanje kakovosti in znižanje cene proizvodov;
- Preglednost medsebojne odvisnosti zmanjša verjetnost napak in omogoča hitrejše izvajanje sprememb.

V podjetju ITS intertrade se ukvarjamo z uvajanjem, načrtovanjem in prilagajanjem informacijskega sistema BAAN v podjetja. Uvajanje izvajajo svetovalci, ki pa rabijo

tudi dobro programsko in sistemsko podporo. Kljub veliki prilagodljivosti programskega paketa pride zaradi narave poslovanja podjetja velikokrat do potrebe po prilagoditvi dela programskega paketa. Svetovalci so zadolženi za identifikacijo in rešitev problema (torej kakšno prilagoditev je potrebno izvesti), samo prilagoditev pa izvajajo razvijalci – programerji. Ko je prilagoditev narejena, jo je mogoče ponovno uporabiti oziroma hitro prilagoditi novim, podobnim potrebam. To pomeni, da je dobro poznavanje že izvedenih prilagoditev velikega pomena za podjetje.

V preteklosti potek razvijanja prilagoditve sistema ni bil dokumentiran. Svetovalec je ustno naročil razvijalcem, kaj je potrebno spremeniti in na kakšen način. Razvijalec je velikokrat prilagoditev izvedel tudi pri stranki. To pomeni, da v podjetju prilagoditev ni bila dokumentirana.

Posledica takšnega večletnega dela je, da se je za premnogimi prilagoditvami izgubila vsaka sled. To pomeni, da je prihajalo do primerov, ko je bila prilagoditev izvajana večkrat oziroma bi bila lahko s pomočjo prejšnje prilagoditve izvedena mnogo hitreje. Prav tako bi bilo mogoče marsikatero implementacijo informacijskega sistema izvesti na lažji, lepši in uporabnikom prijaznejši način, če bi imeli svetovalci dostop do podrobne dokumentacije o že obstoječih prilagoditvah, ki bi jih lahko uporabili. To bi omogočilo hitrejše in uspešnejše uvajanje informacijske sistema in s tem tudi večje zadovoljstvo strank.

Namen magistrskega dela je spoznati način dela v razvojnem okolju podjetja, identificirati problematiko in predlagati rešitev s pomočjo PDM sistema.

Cilj magistrske naloge je zasnova informacijskega sistema za upravljanje z razvojnimi podatki in dokumentacijo izdelkov v podjetju ITS intertrade sistemi, ki bo osnova za uspešno implementacijo sistema. Podrobnejši cilji magistrskega dela so naslednji:

- analiza obstoječega stanja dela v razvojnem oddelku podjetja, definiranje uporabnikov, postopkov ter spoznati informacijsko infrastrukturo in tehnologije,
- s pomočjo ključnih uporabnikov definirati funkcionalnost načrtovanega informacijskega sistema v razvojnem okolju podjetja in področja uporabe, ki jih bo sistem pokrival,
- glede na postavljene zahteve in tehnologije, ki so na voljo, predlagati module PDM sistema, ki se jih bo uporabilo ob implementaciji PDM sistema,

- pripraviti nadaljnji načrt razvoja in optimizacije PDM sistema v podjetju ITS intertrade sistemi.

Predvidena rešitev je načrt implementacije osnovnih funkcionalnosti PDM sistema v razvojnem oddelku podjetja in predvidene nadaljnje možnosti razširitve uporabe sistema (povezava z ERP sistemom Baan, ki ga uporablja podjetje, razširitev uporabe sistema na druga področja v podjetju (nabava, prodaja...)).

V prvem delu magistrske naloge sem opisal PDM sisteme. Opisal sem teorijo PLM filozofije, namen in področja uporabe PDM sistemov, prednosti, ki jih prinaša uporaba takšnega sistema v posameznih delih podjetja. Opisal sem tudi načine, kako z uporabo PDM sistema dosežemo želene cilje.

V drugem delu magistrske naloge sem opisal korake pri načrtovanju in implementaciji informacijskega sistema za podporo upravljanja s podatki in dokumentacijo izdelkov skozi njihovo življenjsko dobo. Najprej sem prikazal analizo obstoječega sistema in opis analize zahtev in funkcionalnosti bodočega sistema. Analizo obstoječega sistema in načrt funkcionalnosti uvedenega PDM sistema sem naredil s pomočjo priprave UML diagramov.

Metoda dela, ki sem jo uporabil, je sestavljena iz priprave UML diagramov na podlagi analize stanja v podjetju in intervjuja z načrtovanimi uporabniki sistema in priprave načrta za nastavitev pilotskega modela sistema (prototipnega modela). UML diagrami so služili za osnovo pri klasični pripravi pilotskega modela sistema, ki ga uporabljamo v podjetju pri uvajanju PDM sistema v druga podjetja. Pri delu sem ugotavljal smiselnost uporabe UML jezika oziroma UML diagramov v procesu nastavitve pilotskega modela PDM sistema. Ugotavljal sem, kateri diagrami prinesejo največ koristi pri pripravi načrta uvajanja in kateri nam ne prinesejo zadosti prednosti, da bi lahko upravičili dodaten strošek pri uvajanju sistema.

Pri izdelavi magistrske naloge sem poskušal združiti moje dosedanje izkušnje s področja informatike (uvajanje PDM in ERP sistemov) z vsebinskim področjem procesov v razvoju podjetja ITS intertrade sistemi. Pri tem sem upošteval smernice, ki jih narekuje razvoj PDM sistemov. Hkrati sem naredil zasnovo načrta implementacije PDM sistema odprto in tako, da bo omogočena nadaljnja gradnja in dopolnjevanje funkcionalnosti ter razširitev na druga področja v podjetju (prodaja, nabava...).

V tretjem delu magistrske naloge sem pripravil zasnovo za kasnejši razvoj sistema na druga področja v podjetju. Pripravil sem idejno zasnovo uporabe ERP sistema

Baan za vodenje projektov uvajanja ERP in PDM sistemov (v oddelku Baan svetovanja) pri drugih strankah in možnost povezovanja s PDM sistemom, katerega uporaba je bolj natančno definirana in v kratkoročnem planu uporabe v podjetju ITS. Možnost uporabe projektnega modula ERP sistema Baan je idejna zasnova, katero bo potrebno kasneje prilagoditi zahtevam, ki se bodo pokazale z delom v PDM sistemu in je zato v dolgoročnem planu uvajanja v podjetje.

V sklepnem delu sem povzel ugotovitve, do katerih sem prišel pri pripravi načrta implementacije PDM sistema v razvojnem oddelku podjetja ITS intertrade sistemi.

2. PDM-sistemi

2.1 Mesto in vloga PDM-sistema v podjetju

Planiranje in spremljanje proizvodnje imata velik pomen v serijski proizvodnji. PDM Sistem in ERP Sistem (Enterprise Resource Planning) morata biti tesno povezana v zaključeno celoto [Downs, 1992, str. 24]. Pri takšnem informacijskem sistemu je pomemben še pregled nad dokumenti v času nastajanja in v času uporabe ter kontroliran dostop do njih v celotnem podjetju. To je tudi v skladu s standardi o zagotavljanju kakovosti. Izredno velik pomen ima učinkovito obvladovanje sprememb [Duhovnik, 2000, str. 1.7].

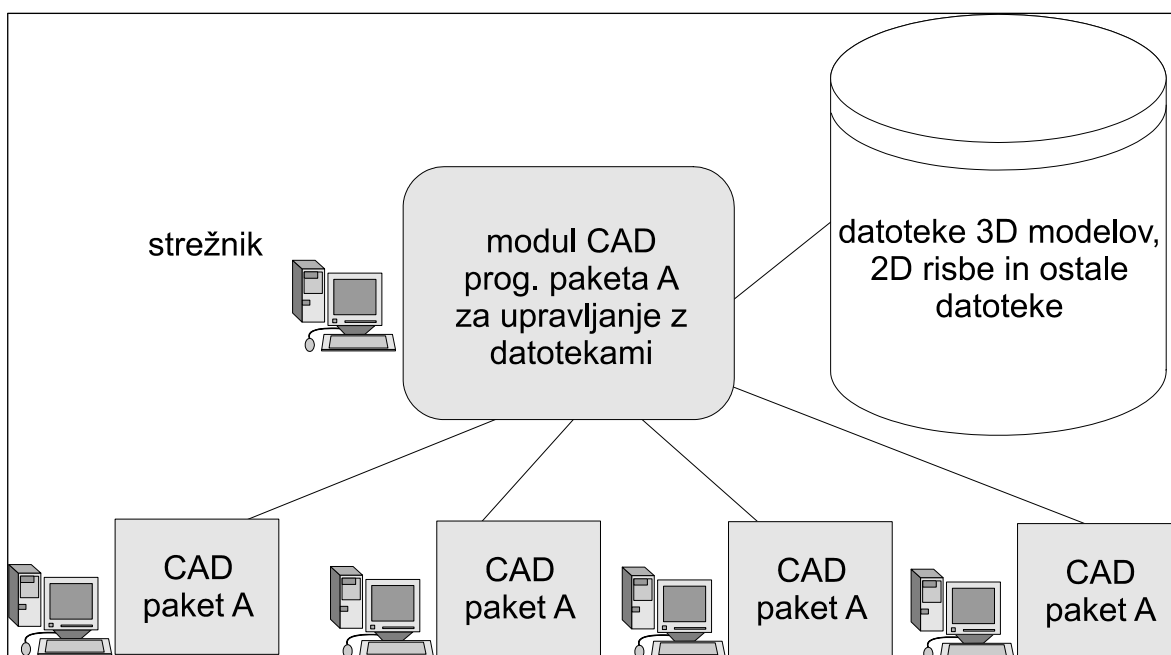
Glede na različne potrebe podjetij poznamo tri osnovne oblike informacijskega sistema upravljanja s podatki izdelkov:

1. Podpora razvojni skupini pri konstruiranju izdelka in kreiranju dokumentov

V kolikor izdelek razvija mešana razvojna skupina, potrebujemo sledenje različnih verzij in kontrolo dostopa do posameznih gradnikov in dokumentov. V takšnem okolju se pogosto spreminjajo. Informacijski sistem mora biti tesno integriran s CAD orodji (Computer Aided Design). Enostavne PDM sisteme lahko najdemo v okviru CAD programskih paketov. Ti pomagajo upravljati z datotekami risb, 3D modeli ipd. Obstajajo kot moduli v CAD programskih paketih ali pa so se razvili v samostojne PDM sisteme [French, 1985, str. 66].

V razvojnem oddelku podjetja je potrebno pregledno hraniti tehnično znanje podjetja in imeti pregled nad različnimi verzijami izdelkov. Dokumentirane morajo biti posamezne odločitve in zgodovina razvoja. Potreben je pregled nad vsemi gradniki, ki so se izdelovali.

SLIKA 1: Enostaven PDM sistem v okviru CAD programskega paketa

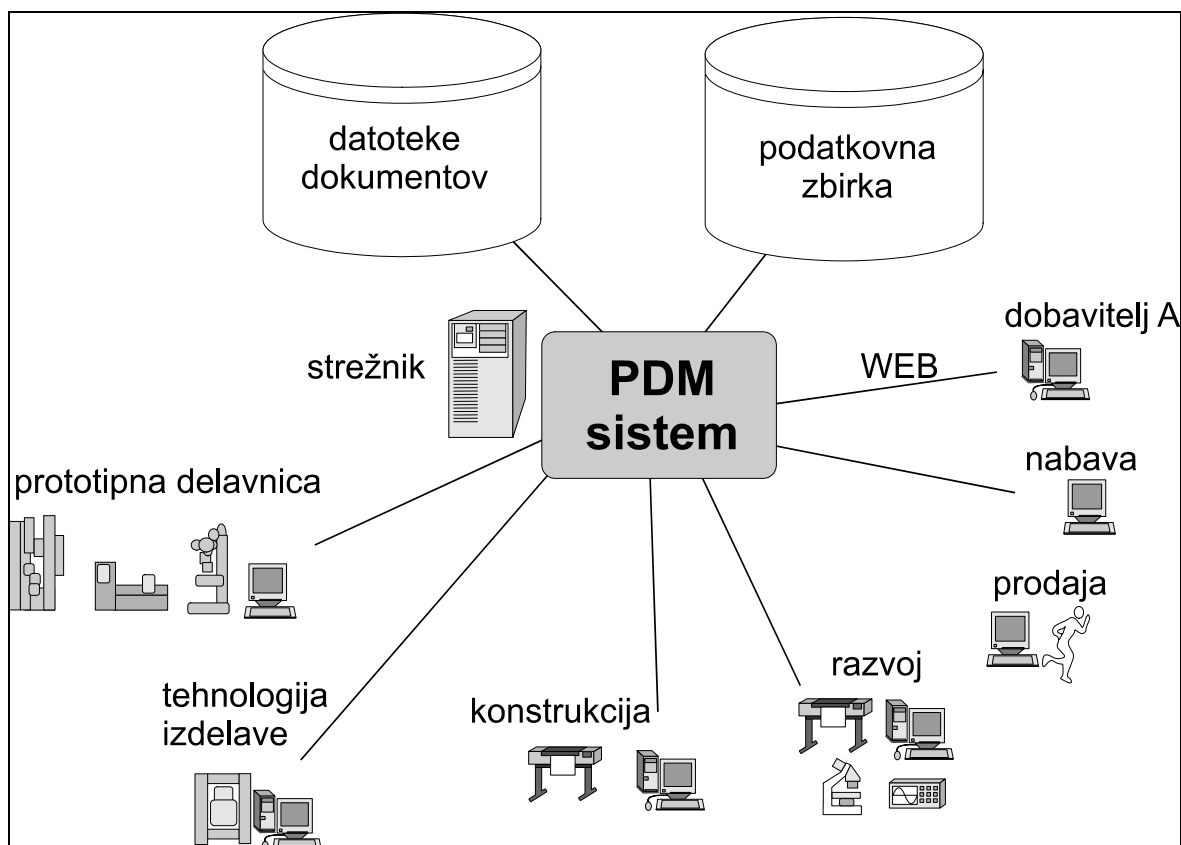


Programska oprema za upravljanje z datotekami znotraj CAD programov je praviloma manj zahtevna za vzdrževanje in cenejša od PDM sistemov. Potrebno je poiskati kompromis med univerzalnimi rešitvami, ki jih ponujajo PDM sistemi in orodji znotraj CAD paketov. V PDM sistemih so podatki bolj strukturirani in zato stabilnejši.

2. Upravljanje podatkov v podjetju s pomočjo samostojnega PDM sistema

Za razliko od prvega primera, kjer je CAD sistem dopolnjen z modulom za upravljanje z datotekami, je to značilna oblika PDM sistema. PDM sistem deluje kot okolje za podporo pri nastajanju in hranjenju dokumentacije v različnih delih podjetja. Omogoča dostop do podatkov širšemu krogu uporabnikov iz celega podjetja in je vezni člen med prostorsko razpršenimi oddelki [Gaus, 1995, str. 18]. V njem so dostopni vsi podatki o izdelkih (3D modeli, delavniške risbe, tekstovni dokumenti...). Lahko vsebuje tudi podatke o različnih notranjih ali zunanjih projektih, ki potekajo v podjetju, kakor tudi celotno nabavno, prodajno in drugo dokumentacijo.

SLIKA 2: Samostojen PDM sistem v podjetju



Tesno povezavo med CAD in PDM sistemom dosežemo s pomočjo CAD integracij v okviru PDM sistema. Te integracije običajno omogočajo dostop do podatkov v PDM sistemu s pomočjo dodatnega menija v CAD orodju. Integracija mora prav tako omogočati avtomatski prenos podatkov med obema sistemoma (kosovnice, nazivi izdelkov, številke rib...). Univerzalna uporabnost PDM sistema in stabilnost skozi čas daje trden okvir, podoben poslovno proizvodnemu sistemu (ERP). PDM sistem pomaga obvladovati zahtevno strukturo in količino podatkov v času njenega nastajanja in kasneje pri konfiguriranju izdelkov za različne potrebe kupcev [Duhovnik, 2000, str. 1.9]. PDM sistem ima pomembno vlogo kot orodje za komunikacijo in obvladovanje sprememb v podjetju.

3. Hranjenje podatkov o izdelku v poslovno informacijskem sistemu (ERP)

V podjetjih, ki se ukvarjajo s pretežno proizvodno dejavnostjo je nujno potrebna povezanost sistema za upravljanje z razvojnimi podatki in dokumentacijo (PDM sistem) s poslovno proizvodnim informacijskim sistemom (ERP sistem). V ERP

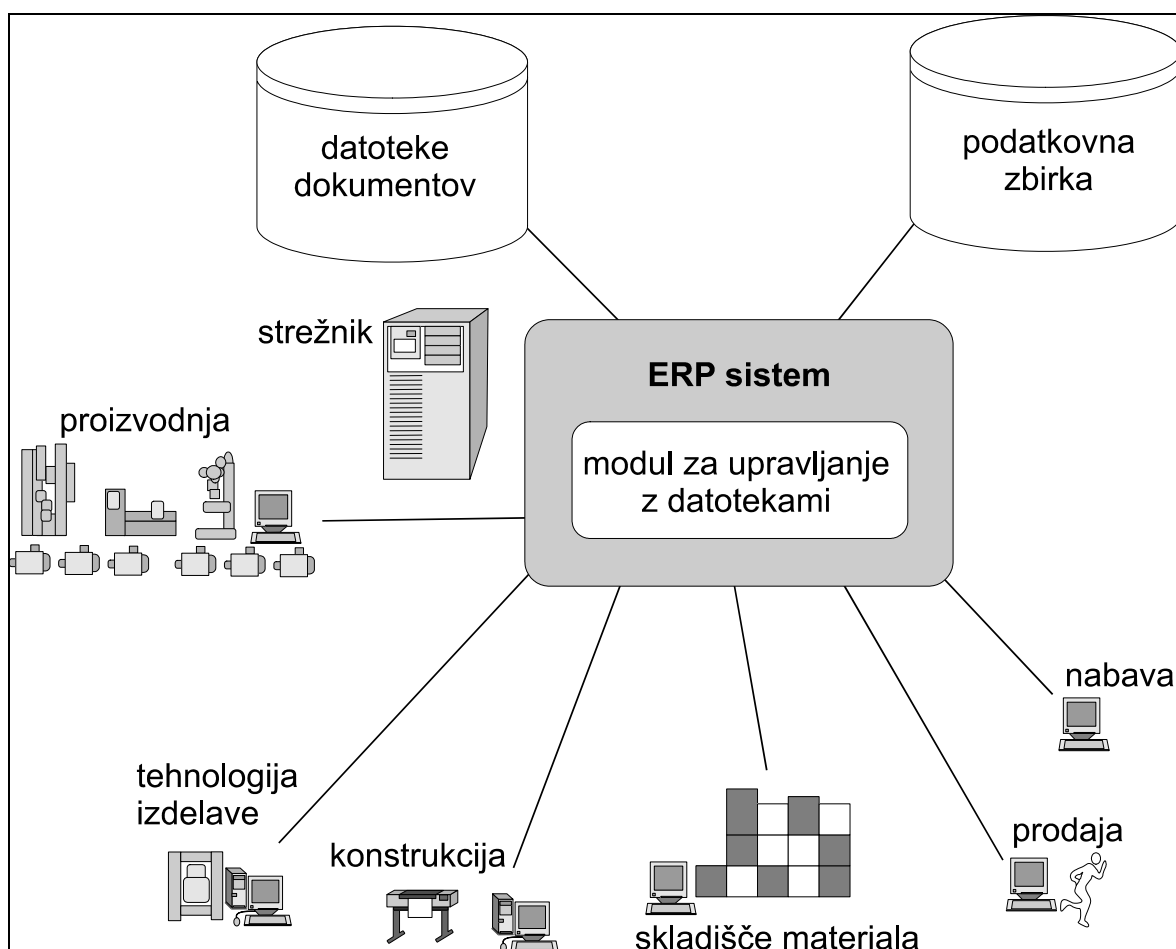
sistemu se upravljajo podatki, ki so veljavni in se ne spreminjajo pogosto. Pomembno vlogo ima obvladovanje sprememb, ki omogoča doseganje standardov kakovosti. V takih sistemih se razvojno konstrukcijski proces obravnava ločeno. V ERP sistem se ponavadi prenese samo veljavni in končni del rezultata tega procesa. Podatki so prilagojeni potrebam povprečnega uporabnika. V oddelkih nabave, prodaje, proizvodnje... praviloma zadostujejo vpogledi v dokumentacijo in ne potrebujejo CAD orodij za pregledovanje [Fripp, 1993, str. 12].

Za opisano vlogo je najbolj primerna razširitev funkcionalnosti ERP sistema z upravljanjem z dokumenti. Problem se praviloma pojavi pri dveh ključnih delih informacijskih tokov. Problemi nastajajo predvsem zaradi drugačne zasnove dela v razvojno konstrukcijskem procesu v primerjavi z obdelavo v proizvodno poslovnem procesu [Duhovnik, 2000, str. 1.10]:

- Prevelika dostopnost koristnikov informacij do podatkov v razvojni fazi izdelka, kar lahko preprečimo s pomočjo statusov podatkov in avtorizacijo.
- Preslikava podatkov iz razvojno konstrukcijskega procesa v ostali del proizvodnega procesa, kar lahko rešimo s pomočjo omejitev pri preslikavi.

Razvojna skupina ima drugačne zahteve kot proizvodni sistem. Proizvodna kosovnica je drugačna kot razvojna. Ima druge materiale, polizdelke... Direktna preslikava ponavadi ni mogoča brez dopolnitev. Možno je ločeno shranjevanje proizvodne in razvojne kosovnice, pri čemer je potrebno zagotoviti sprotno usklajevanje ob spremembah, ali pa izvesti neposredno integracijo med obema, pri čemer jih je potrebno prilagoditi tako, da izhajata ena iz druge.

SLIKA 3: PDM sistem v povezavi z ERP sistemom.



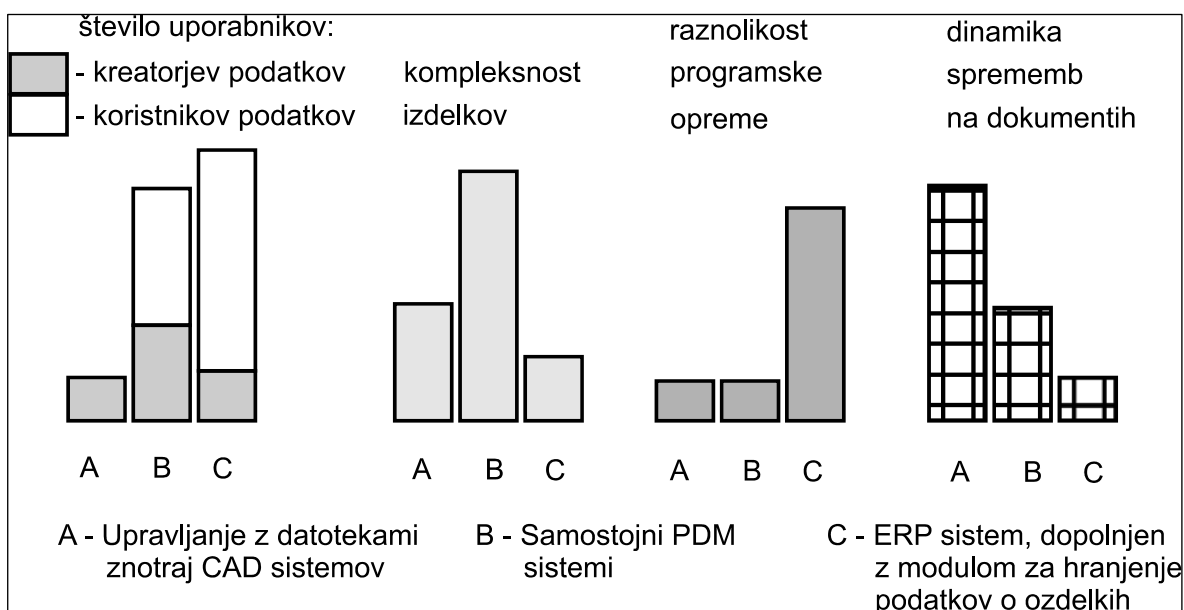
Obstajajo tudi PDM sistemi, ki lahko delujejo samostojno ali v povezavi z več ERP sistemi. Te sisteme lahko začnemo uporabljati samostojno in jih kasneje nadgradimo s povezavo z ERP sistemom. V teh primerih skrbi za povezavo poseben vmesnik, ki skrbi za prenos podatkov iz PDM sistema v ERP sistem in obratno. Vendar tudi v tem primeru prenašamo v ERP sistem samo podatke, ki jih potrebujemo za poslovanje (torej samo aktivno verzijo). Vsi podatki o sledljivosti in stare verzije ostanejo v PDM sistemu.

Predstavitev različnih dejavnosti glede na tri osnovne oblike informacijskega sistema

Na SLIKI 4 je predstavljena razmejitev vseh treh oblik PDM sistema (vodenje dokumentov znotraj CAD orodja (A), samostojen PDM sistem (B) in ERP sistem z razširjeno funkcionalnostjo (C)) [Duhovnik, 2000, str. 1.11]. V A in C primeru morata CAD orodje in ERP sistem imeti možnost razširjene funkcionalnosti za

hranjenje datotek. Upoštevani so kriteriji števila uporabnikov, kompleksnosti izdelkov, raznolikosti programske opreme, hranjenje datotek in dinamika sprememb. Pomemben vpliv imajo seveda tudi značilnosti posameznih podjetij, vendar v grobem velja naslednje. V okolju CAD sistema deluje manjše število uporabnikov, ki si med seboj zaupajo. V tem primeru lahko zaradi pogostih sprememb poenostavimo dostop in možnost spreminjanja [Perlitz, 1996, str. 142]. Vsi uporabniki morajo uporabljati enotno programsko opremo. PDM sistem pri večjem številu uporabnikov omogoča nadzor in večjo preglednost tudi pri bolj kompleksnih izdelkih [Hubka, 1988, str. 51]. Takšen sistem je bolj univerzalen in omogoča uporabo različnih programskih orodij, kar pa zahteva več časa za pripravo podatkov. V primeru uporabe ERP sistema z razširjeno funkcionalnostjo pa v ERP sistemu hranimo le končne rezultate, do katerih lahko dostopa večje število uporabnikov. Za dostop do podatkov se uporablja enotna programska oprema, medtem ko nastajanje dokumentacije ni podprto. S takšnim sistemom se praviloma tudi zmanjša dinamika sistema [Jones, 1997, str. 34].

SLIKA 4: Odvisnost obsega dejavnosti od načina vodenja podatkov o izdelkih.



Novejši PDM sistemi pokrivajo delovanje samostojnega PDM sistema in tudi komunicirajo z ERP sistemi. Skrbijo za nastajanje dokumentacije in v ERP sistem pošiljajo končne rezultate od koder so dostopni uporabnikom ERP sistema. Takšen sistem zahteva nekaj več časa in stroškov v toku uvajanja (predvsem pri nastavitvi komunikacije z ERP sistemom), vendar združuje oba sistema, zmanjšuje možnosti napak pri vnosih in spremembah in s tem podpira celotno poslovanje podjetja [Kalpič, 1998, str. 23].

PDM sistem naj bi pokrival le funkcionalnost, ki jo določeno okolje potrebuje, saj postavitve kompleksnega PDM sistema zahteva dražje vzdrževanje in zahtevnejše uvajanje [Prasad, 1997, str. 182]. To pomeni, da manjša razvojna skupina, ki tesno sodeluje, PDM sistema ne potrebuje za medsebojno komunikacijo. Potrebuje predvsem učinkovit sistem arhiviranja podatkov in komunikacijo s CAD orodji [Prasad, 1996, str. 356]. Na drugi strani pa lahko učinkovito in pregledno delo večje skupine dosežemo le s PDM sistemom, ki pa je višjega cenovnega razreda. V tem primeru pa moramo zagotoviti ustrezno nastavitve sistema, vzdrževanje in izobraževanje uporabnikov [Perlit, 1996, str. 92].

V primeru novejših PDM sistemov, kjer končne rezultate avtomatsko prenašamo v ERP sistem, pa dosežemo tudi znatno zmanjšanje možnosti napak pri podvojenih vnosih v ločen ERP sistem. Takšne sisteme je smotrno uporabiti v okrnjeni obliki tudi v primeru manjše razvojne skupine, ki pa jo dopolnjuje tehnološka skupina, ki deluje v ERP sistemu.

2.2 Povezovanje med PDM in ERP sistemi

Če želimo razumeti vlogo PDM in ERP sistemov, moramo poznati temeljne razlike in podobnosti med proizvodnjo in razvojem izdelkov. Na proizvodnjo in razvoj lahko gledamo podobno, le da je pri razvoju rezultat dokumentacija, medtem ko je v primeru proizvodnje rezultat materialni izdelek. V obeh primerih imamo ljudi in opremo, ki morajo biti čimbolj učinkoviti in prinašati dobiček. Strošek razvoja je v računovodstvu in kontrolingu ponavadi prikazan zamegljeno in ne tako direktno kot strošek proizvodnje, kjer se pazi na vsako minuto. Posledica je, da se ne razmišlja toliko o pomenu in vlaganju v izboljšave v razvojnih korakih [Reinertsen, 1997, str. 155]. Prav razumevanje jasnih razvojnih smeri in stroškov postaja kritična razlika v prodornosti posameznih podjetij. Zakrnel razvojni oddelk lahko tako predstavlja temeljno blokado v razvoju podjetja. V tem primeru je najbolj primerna stroškovna opredelitev rezultatov razvojnega dela [Duhovnik, 2000, str. 1.12].

V proizvodnji imamo pred pričetkom dela na voljo vse informacije, kar pa ne drži za razvoj izdelkov. V času razvoja izdelka zbiramo in odkrivamo nove informacije, ki imajo pogosto odločilen vpliv na potek dela. Delo je odvisno od mnogih stranskih učinkov in ga je zato težko natančno načrtovati. Razvoj izdelka se ponavadi izvaja enkrat s številnimi različnimi možnostmi in se ne ponavlja kot je to primer v proizvodnji izdelkov. V razvoju se nenehno iščejo nove izboljšave in dopolnitve in pogosto časovni pritiski povzročijo zaključek razvojnega procesa [Grabowski, 2001, str. 112].

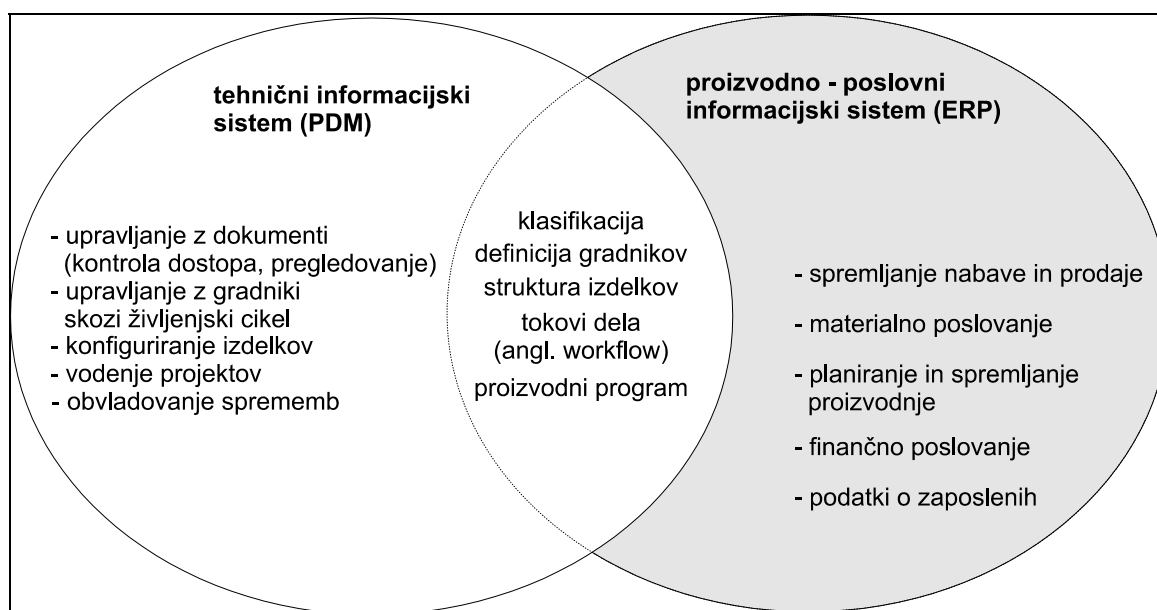
Očitno je, da so bili PDM sistemi razviti na osnovi precej drugačnih zahtev kot proizvodni informacijski sistemi (ERP sistemi). Odpira pa se vprašanje ali je mogoče funkcionalnosti ERP sistemov uporabiti tudi za potrebe razvoja.

Glavni del ERP sistema je baza podatkov o izdelkih z vsemi osnovnimi podatki. Jedro ERP sistema so kosovnice različnih vrst (modularna, strukturna, proizvodna, variantna...). ERP sistemi ponavadi omogočajo spremljanje materialnega poslovanja (skladiščenja, nabave, prodaje) in proizvodnje. Stopnja podpore planiranju je lahko pri različnih sistemih zelo različna [Kerbel, 1993, str. 233]. V sklopu ERP sistema je ponavadi zajeto tudi finančno in kadrovske področje.

Uporaba PDM sistemov je bila v preteklosti omejena na razvojne oddelke za podporo pri shranjevanju in dostopu do datotek, ki nastajajo pri računalniško podprtem konstruiranju. Zato so najprej na trg prišli PDM sistemi kot moduli CAD orodij [Koller, 1994, str. 277]. Samostojni, od CAD sistemov neodvisni sistemi, so sledili kasneje. Danes se področje delovanja PDM sistemov širi na celotno podjetje. PDM sistem je postal orodje za upravljanje s podatki o izdelkih skozi celotno življenjsko dobo. Zato razširjene PDM sisteme v zadnjem času imenujemo tudi PLM sistemi (Product Lifecycle Management). Lokacijske meje podjetja niso več meje informacijskih tokov. Osnovne funkcionalnosti PDM sistemov so spremljanje in vodenje procesa konstruiranja in nadzor nad spremembami (sledljivost), upravljanje s strukturo proizvodov, klasifikacija in projektno vodenje. Marsikateri PDM sistemi se lahko uporabijo tudi kot sistem za vodenje potrjevanja in arhiviranja druge dokumentacije (npr. za arhiviranje in potrjevanje ponudb v prodajnem oddelku).

Hitro lahko ugotovimo, da se področji PDM in ERP sistemov v marsičem prekrivata (npr. struktura proizvoda). Nujno je poznavanje prednosti obeh sistemov in možnost njihove povezave v učinkovit sistem [Bartuli, 1995, str. 3]. Prav tako razvijalci vedno pogosteje potrebujejo pri svojem delu informacije iz ERP sistema (podatki o dobaviteljih, dobavnih rokih, razpoložljivih zalogah materialov in njihovih cenah). V tem primeru je smiselno zagotoviti tudi podatke iz ERP sistema v PDM okolju, saj je brskanje podatkov iz dveh različnih sistemov zamudno in neučinkovito. Razvijalci delajo pretežno v PDM sistemu in rabijo samo omejeno količino podatkov iz ERP sistema, ki jih v primeru integriranega in enotnega sistema lahko zagotovimo na enem mestu.

SLIKA 5: Funkcionalnost PDM in ERP sistemov in njuno prekrivanje



Celovito upravljanje s komponentami izdelkov je ponavadi bolj razvito v ERP sistemih, medtem ko so grafično podprte metode dela (workflow) v PDM sistemih mnogo bolj razvite. Dolgoročno bosta PDM in ERP sistem vse bolj integrirana ali pa se bosta zlila v enoten sistem. V prehodnem obdobju pa je potrebno najti ustrezno povezavo obeh sistemov. Čim višja je stopnja integracije, več podatkov se prenaša med obema sistemoma, kar pomeni določeno podvajanje [Grabowski, 1998, str. 55]. Odločiti se je potrebno, kateri sistem je prevladujoč in kako vsebuje izhodiščne podatke, ki se prenašajo naprej. Na SLIKI 6 je narisanih več možnosti in vsaka od njih je lahko uspešna [Duhovnik, 2000, str. 1.14].

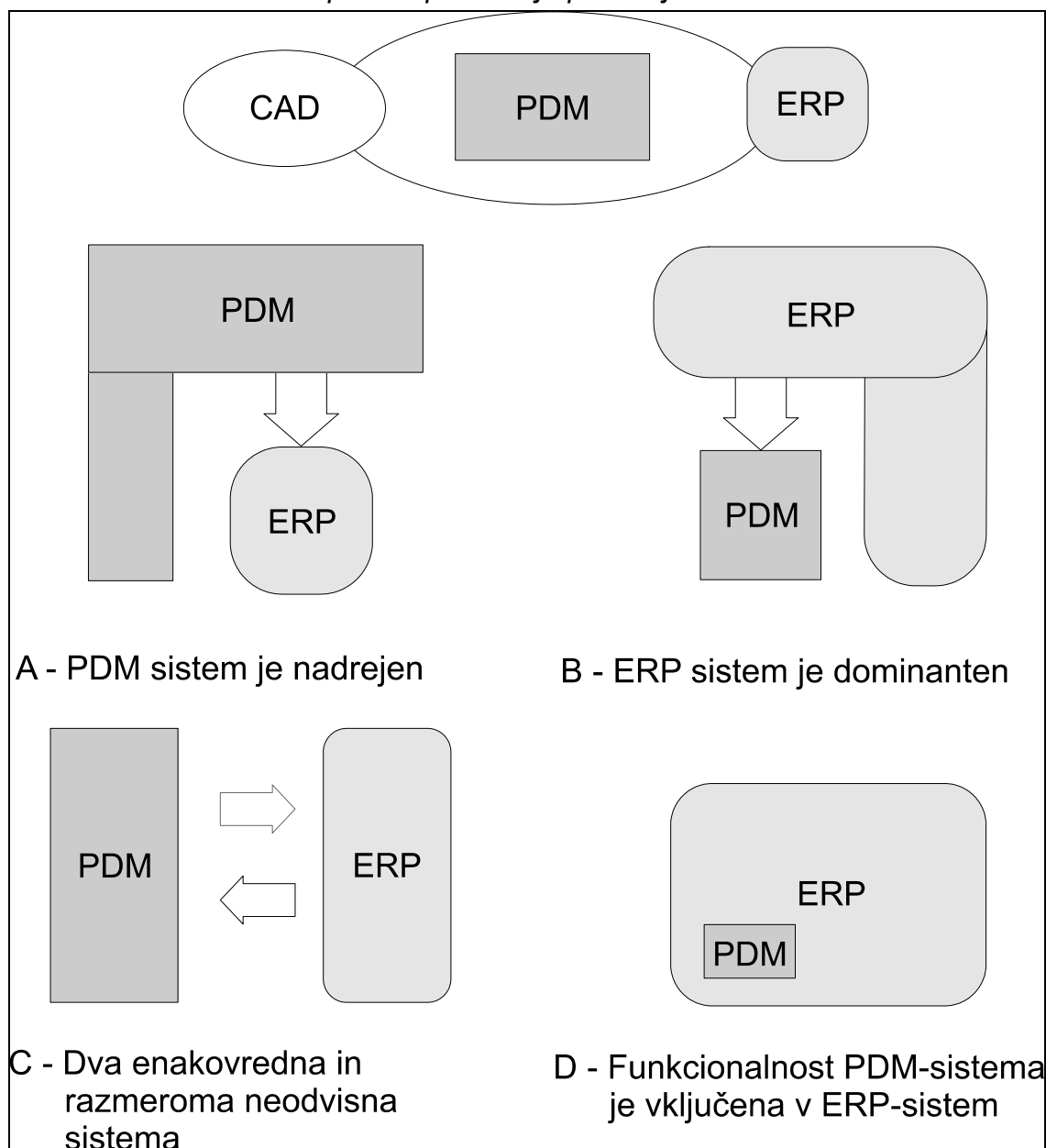
V primeru A ima PDM sistem poglavitno vlogo. V njemu se kreirajo podatki in prenašajo v ERP sistem. Za mnoga podjetja (še posebej za posamično in maloserijsko proizvodnjo) je podatek o izdelku ključnega pomena. Informacije in dokumentacija nastajajo v tem zaporedju in je zato takšna pot najbolj naravna.

Primer B nastane kadar ima podjetje vpeljan ERP sistem pred PDM sistemom. Glede na tradicijo ERP sistemov so ti primeri zelo pogosti. ERP sistem je prevladujoč in proizvodnja zahteva prilagojeno strukturo podatkov. Morebitni izvedeni popravki v času izdelave se prenašajo tudi v PDM sistem. Takšen način je primeren, kadar popravki ne posegajo v tehnično dokumentacijo in se izvedejo le dopolnitve.

V C primeru gre za dva neodvisna sistema, kar je pogosto primer v masovni proizvodnji. Podatki se šele v zreli fazi razvoja prenesejo v proizvodnjo in zato ni potrebe po veliki izmenjavi podatkov.

D primer ponazarja primer ERP sistema, ki ima razširjene funkcionalnosti na področje PDM sistema. Funkcionalnosti takšnih sistemov so ponavadi omejene na hranjenje že potrjene dokumentacije in omogočajo dostop do podatkov širšemu krogu uporabnikov. Ta rešitev je primerna v proizvodnih podjetjih brez izrazite razvojne funkcije.

SLIKA 6: Različni primeri pokrivanja področij PDM in ERP sistema.



Uvajanje višje stopnje integracije med obema sistemoma lahko podjetju omogoči doseganje standardov kakovosti (ISO). Elektronsko vodenje dokumentacije lahko izvedemo na različnih nivojih v podjetju. Za doseganje tega cilja so ponavadi potrebne organizacijske spremembe. Ročno podpisovanje in potrjevanje dokumentacije mora nadomestiti elektronsko. Razvoj novih tehnologij ima močan vpliv na organizacijo v podjetju, zato vpeljevanje višje stopnje integracije brez organizacijskih sprememb in optimizacije procesov pogosto ni smiselno. Takšna prenova je zahteven proces, kjer se morajo nujno upoštevati posebnosti posameznega podjetja.

2.3 Zagotavljanje kakovosti (ISO) in informacijski sistem

Razvoj in uporaba sistemov kakovosti sta se pričela v ameriški vojaški industriji pri vodenju obsežnih projektov s številnimi podizvajalci. Upoštevanje zahtev standarda pripomore k postavitvi preglednega in učinkovitega sistema [Duncan, 1990, str. 116]. Sistemi kakovosti postavljajo zahteve glede sledljivosti dokumentov, potrjevanja odgovornih oseb, zagotovljeni morajo biti ustrezni postopki ob spremembah, zahtevano je sodelovanje v prvih razvojnih fazah. Sistemi kakovosti zahtevajo minimalne pogoje za doseganje kakovosti. Način zagotavljanja postavljenih zahtev standarda s standardom ni predpisan. Dokumenti so lahko v elektronski ali papirnati obliki. PDM sistemi z vgrajenimi sistemi potrjevanja dokumentacije in mehanizmi kontrola dostopa do dokumentacije v elektronski obliki so veljavni načini zagotavljanja standarda kakovosti. Sistem mora biti zasnovan tako, da so zahteve standarda vgrajene vanj. Računalniška podpora poenostavi izvajanje standarda o zagotavljanju kakovosti, za uporabnika je prijaznejša, zmanjša se možnost napak in količina papirja. Uporabnik je pri delu voden in ima kontroliran dostop do informacij [Duhovnik, 2000, str. 1.16].

PDM sistem zaradi popolne sledljivosti omogoča nastanek virtualnih razvojnih skupin, kjer člani skupine lahko spremljajo nastajanje izdelkov in dokumentov. Potrjevanje dokumentov se izvaja preko elektronskih podpisov oziroma sprememb statusov dokumentov. Izvajanje sprememb je hitro, saj pregled lahko izvaja več oseb hkrati. Sledljivost dokumentov je lokacijsko neodvisna. Papirnati dokumenti se pojavljajo le še kot pomožni nosilec informacij. Uporabniki imajo v informacijskem sistemu vedno zagotovljen dostop do zadnjih in pravih informacij.

Postavitev sistema kakovosti je v veliki meri odvisna od velikosti in dejavnosti podjetja. V družinskem podjetju manjšega obsega je lahko papirnata ureditev dokumentov hitrejša in cenejša od uporabe računalnikov, če se naslonimo na

učinkovitega in zanesljivega posameznika [Duhovnik, 2000, str. 1.16]. Kakor hitro pa količina podatkov naraste in si postavimo dodatne zahteve, kot so odzivnost, ekonomičnost, preglednost, se v večjih podjetjih pokaže potreba po višji stopnji integracije. V primeru nezanesljivega upravljavca z informacijami (dokumenti) pa je uporaba PDM sistema ali drugega informacijskega sistema nujna.

Vsaka urejena organizacija, ki uporablja PDM sistem brez težav zadovoljuje zahtevan kriterij standarda kakovosti. V PDM sistemu so uporabniki razvrščeni v skupine s točno določenimi pooblastili. Točno je določeno katere dokumente lahko kreirajo, spreminjajo, katere lahko le berejo in do katerih sploh nimajo dostopa. Vsak trenutek je dostopna zgodovina nastajanja in spreminjanja dokumentov [Duhovnik, 2000, str. 1.18].

Uspešnost sistema kakovosti je odvisna od doslednosti izvajanja postopkov in navodil. V PDM sistemu je mogoče mnoge postopke avtomatizirati, zato je njihovo izvajanje hitrejše in doslednejše kot s papirnatimi dokumenti. Elektronska oblika dokumentov v PDM sistemu je primerna tudi za hranjenje in vzdrževanje dokumentacije v sistemu kakovosti.

Urejen PDM sistem je tudi zelo primeren za dopolnjevanje in usklajevanje pogodbene dokumentacije med različnimi službami v podjetju. Pogodbena dokumentacija je zelo obsežna in pri pripravi je zelo koristen pregled nad že izvedenimi pogodbami. Možnosti iskanja in povezovanja dokumentov v PDM sistemu sta v veliko pomoč. V primeru, da izmenjava dokumentacije poteka v elektronski obliki, se dopolnitve enostavno dodajajo v PDM sistem. Papirnat dokumente je možno skenirati oziramo označiti mesto hranjenja in tako so vsi dokumenti na razpolago v PDM sistemu.

Informacijski sistem močno olajša pretok informacij in poveča dostopnost in preglednost podatkov v podjetju. V PDM sistemu je vsak trenutek dostopno stanje razvoja izdelka z vsemi pripadajočimi dokumenti. Tokove dokumentov med uporabniki in obveščanje je možno dobro definirati (Workflow). Zagotovi se lahko pregledno sledenje ujemanja poteka razvoja s postavljenim planom [Kos, 1985, str. 20]. V informacijski sistem vgradimo kontrolne točke in zahtevane potrditve. PDM sistem zagotovi izvajanje vseh formalnih zahtev pri razvoju in obenem zaradi elektronske oblike dokumentov celoten postopek močno pospeši.

PDM sistem omogoča dokumentiranje poteka razvoja, izboljša komunikacijo med člani skupina in olajša vrednotenje izpolnjevanja zahtev. V PDM sistemu je mogoče pregledno slediti razvoju. Za vsak gradnik in dokument se ve, v kateri fazi

življenjskega cikla se nahaja in kakšne so pravice dostopa uporabnikov. Za vsako fazo se definirajo pogoji prehoda (kdo mora potrditi izvedeni pregled) [McIntosh, 1995, str. 41].

Osnovna naloga PDM sistema je upravljanje z vsemi dokumenti o izdelkih. Sem spada hranjenje, nastajanje in izdaja dokumentov. V fazi razvoja, ko se dokumenti pogosto spreminjajo, ima elektronska oblika dokumentov prednost. Za vse uporabnike so dokumenti hranjeni le na enem mestu in ob spremembi je potrebno zamenjati le ta original. Uporabniki vedno dostopajo do trenutno veljavne verzije dokumentov. Pregled nad verzijami, dostop do referenčnih dokumentov ter možnost hitre komunikacije v PDM sistemu obravnavo sprememb pospeši in zmanjša možnost napak.

2.4 Osnovna funkcionalnost PDM sistemov

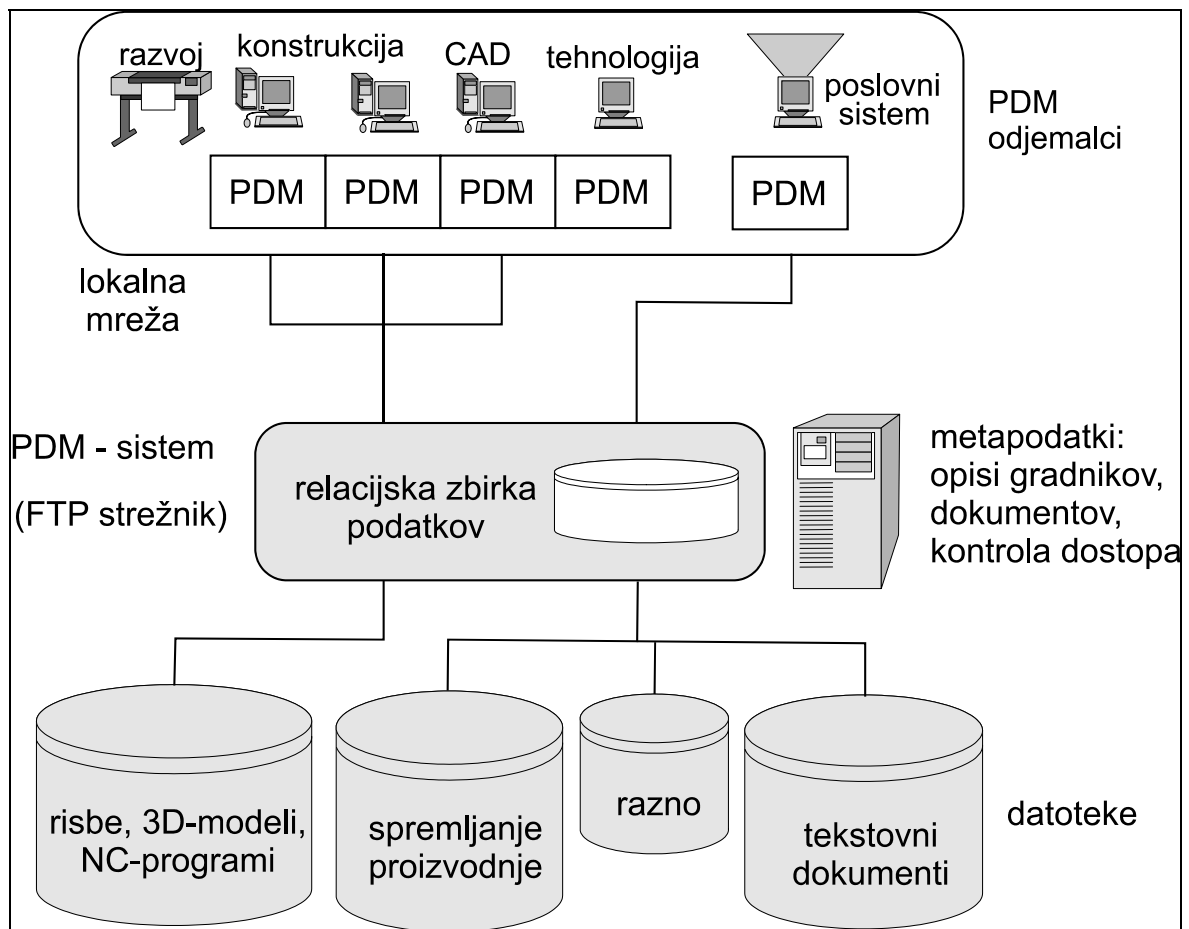
Funkcionalnost PDM sistema temelji na opredelitvi vodenja in pretoka podatkov o izdelkih. Prvi korak pri upravljanju podatkov je sistematična analiza pretvorbe podatkov po količini in vsebini iz obstoječega stanja v elektronsko obliko, ki je primerna za vnos v sistem.

PDM sistem je vmesni člen med posameznimi delovnimi mesti in centralnim arhivom datotek. PDM sistem kontrolira dostop do podatkov in preko opisa (metapodatkov) omogoča iskanje na različne načine in po različnih kriterijih.

V osnovi gre torej za skupni delovni arhiv oziroma hranjenje dokumentov v elektronski obliki. PDM sistemi so praviloma dopolnjeni z obveščanjem med uporabniki in delovnim potekom (workflow). Omogočajo olajšan dostop do podatkov direktno iz CAD orodij preko posebnih vmesnikov.

Hiter in pregleden dostop do podatkov preko klasifikacije ali strukture izdelka je jedro vsakega PDM sistema. Jedro PDM sistema so ponavadi elementi (sestavni deli in sestavljeni artikli). Nanje vežemo dokumente in ostale podatke [Duhovnik, 2000, str. 3.3].

SLIKA 7: Položaj in vloga PDM sistema



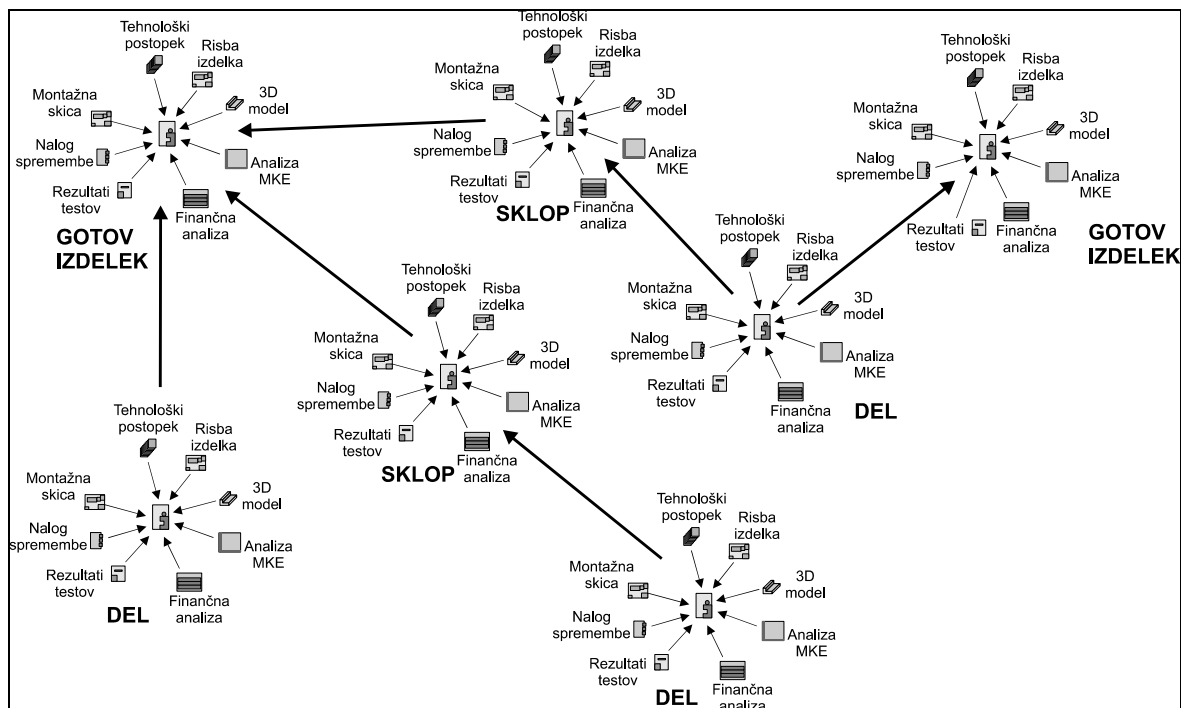
Struktura izdelkov in nanjo pripete informacije

Podatkovni model izdelkov je kompleksen in sestavljen iz množice elementov (podsestavov, materialov) in nanje pripetih dokumentov (SLIKA 8). Elementi, ki sestavljajo izdelek so povezani v različne podsestave in tvorijo osnovno strukturo izdelka (skelet), na katerega so nanizani vsi potrebni podatki [Duhovnik, 2000, str. 3.3]. Dostop do podatkov je mogoč na več načinov. Lahko dostopamo direktno s pomočjo iskanja preko atributov izdelka, s pomočjo iskanja preko strukture izdelka, preko klasifikacijskih atributov ali preko strukture projektnih, arhivskih ali delovnih map. Vsak element v strukturi mora biti neodvisno dokumentiran, da ga lahko uporabimo na različnih mestih. Razlikujemo več vrst kosovnic, ki so prilagojene različnim potrebam uporabnikov [Anderl, 1996, str. 156].

PDM sistem omogoča sprehajanje po strukturi in pri posameznem elementu takojšen dostop do podrobnejšega opisa in pripadajočih dokumentov. V PDM

sistemih z objektnim uporabniškim vmesnikom so elementi z relacijami med seboj povezani v strukturo [Kurbel, 1993, str. 95].

SLIKA 8: Struktura izdelka v PDM sistemu



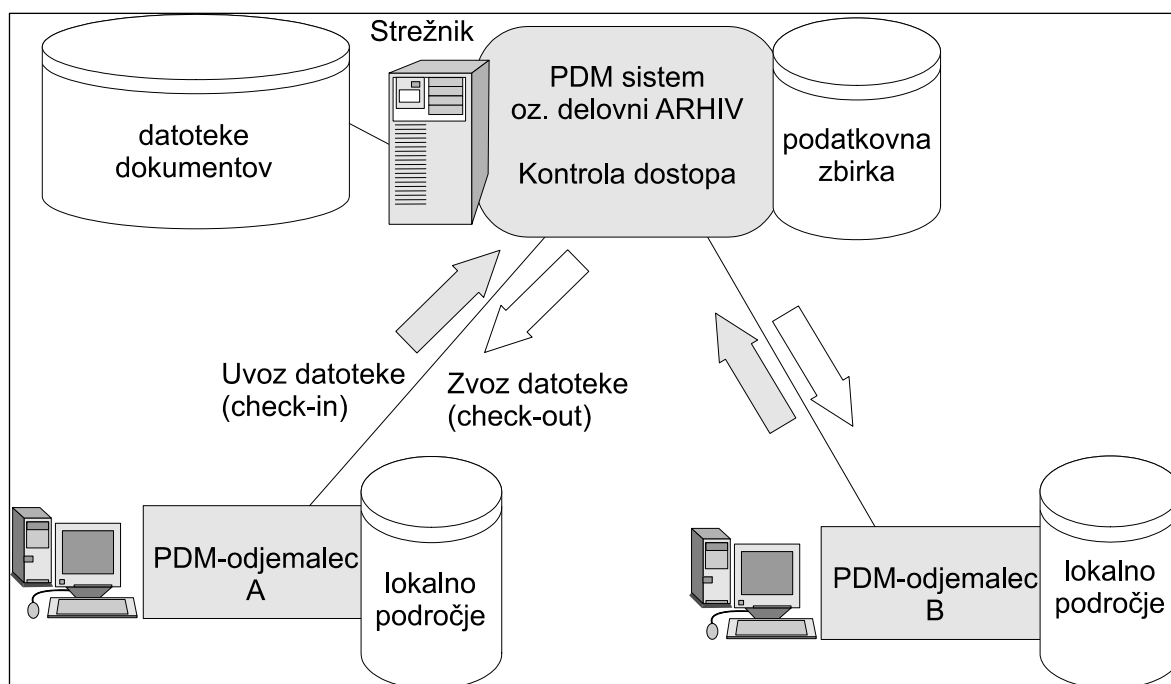
Število nivojev v strukturi je odvisno od kompleksnosti izdelka.

Upravljanje z dokumenti

Sistem za upravljanje z dokumenti omogoča hiter dostop do dokumentov in varno shranjevanje. Omogoča nadzor nad verzijami in zaklepanje podatkov v času spreminjanja.

Podobno kot osnovni elementi (gradniki) so tudi dokumenti opisani z atributi (ident, avtor, datum nastanka, pregleda in potrditve, vrsta dokumenta...). Posamezen dokument je lahko vezan na več osnovnih elementov.

SLIKA 9: Uvoz in izvoz datotek v PDM sistem



Med osnovne operacije pri upravljanju z dokumenti spadata uvoz in izvoz datoteke [Duhovnik, 2000, str. 3.8]. Uvoz datoteke pomeni fizičen transport datoteke od PDM uporabnika na datotečni strežnik PDM sistema. S tem postane datoteka dostopna uporabnikom glede na pravila o dostopu do dokumentov, ki so postavljena v PDM sistemu. Prav tako izvoz datoteke pomeni njen prenos iz PDM strežnika na lokalni disk uporabnika. Ob tem se datoteko v PDM sistemu zaklene, da je zaščitena pred nekontroliranimi spremembami, medtem ko lahko uporabnik svojo kopijo poljubno spreminja. Ta prenos je nujen zaradi normalnega dela z dokumentom, saj je delo preko omrežne povezave znatno počasnejše. Po končanem popravljanju datoteke se ponovno izvede uvoz v PDM sistem, kjer se datoteka hranjena na strežniku posodobi in odklene za uporabo. Odvisno od statusa dokumenta in pravil o sledljivosti sistem dovoli posodobitev datoteke ali pa shrani novo in staro verzijo.

Opredelitev aktivnosti in pravic dostopa uporabnikov

PDM sistemi omogočajo različne vrste uporabe podatkov preko mehanizmov zaščite. Pravice dostopa do podatkov se skozi življenjsko dobo izdelka spreminjajo. Lastnik podatkov lahko svoje dokumente v času nastajanja poljubno spreminja, medtem ko je podatke v arhivu dovoljeno spreminjati le po posebnem, definiranim postopku. V sistemu poznamo osnovne dostope do datotek: branje,

pisanje, brisanje, izvajanje in vidnost datotek, ki so praviloma drugačni za lastnika datoteke, za ožjo skupino, kot za ostale uporabnike [Lipnack, 1997, str. 49]. Pravice so vezane na določeno fazo v življenjski dobi elementa ali dokumenta. Različne možnosti omogočajo številne kombinacije in s tem različne stopnje zaščite. Uporabniki sistema morajo biti posebej določeni v PDM sistemu. Razdeljeni so lahko v delovne skupine, ki jim ločeno dodeljujemo pravice dostopa, kar omogoča lažje vzdrževanje sistema. Poleg pravic, ki jih uporabnik dobi z dodelitvijo v skupino, pa lahko ima vsak uporabnik še dodatne individualne pravice. Dostop do podatkov mora biti prilagojen vlogi uporabnika. Z avtomatizacijo postopkov pretoka, potrjevanja in spreminjanja dokumentov in izdelkov, lahko dosežemo precejšnje časovne prihranke.

Odpiranje novih projektov in potrjevanje dokumentacije je omogočeno le uporabnikom z ustrežno vlogo. Običajen uporabnik pa lahko spreminja le svoje dokumente, ko so ti v fazi nastajanja.

Upravljanje z izdelki in njihovo dokumentacijo skozi življenjsko dobo (Lifecycle management)

V času razvoja je potreben voden pretok dokumentov med uporabniki in pregled nad trenutno fazo obdelave podatkov. Elementom (gradnikom) pripenjamo posamezne dokumente. Vsak dokument se lahko nahaja v različnih fazah svojega življenjskega cikla: v delu, v kontroli, testiranje in podobno. PDM sistem mora zagotavljati primeren dostop glede na trenutni status. Spremembo potrjenih dokumentov lahko izvajamo le s predpisanim postopkom, ki zagotavlja pregledno dokumentiranje spremembe ter pridobitev soglasja s strani prizadetih.

Dodatno preglednost na dokumenti in elementi dosežemo z njihovim povezovanjem v skupine (npr. glede na projekt). Mogoče je izvajati tudi potrjevanje vseh dokumentov v skupini (potrditev celotnega projekta oziroma projektne mape).

Delovni potek (Workflow Management)

Ob povečanju produktivnosti in skrajševanju proizvodnega cikla narašča pretok informacij in dokumentov znotraj podjetja. Papirnati pretok dokumentacije takšen položaj obvladuje z znatnimi stroški (daljši odzivni čas in več ljudi). Veliko pridobimo s shranjevanjem dokumentov centralno v elektronski obliki. Dodatno imamo v PDM sistemu možnost ureditve pretoka dokumentov v elektronski obliki.

PDM sistemi krmilijo pretok podatkov med delovnimi mesti po vnaprej določenem zaporedju. Poskrbljeno je, da se velik del rutinskih nalog izvaja avtomatsko. Uporaba tovrstnih funkcionalnosti še posebno pride do izraza v okoljih, kjer prevladuje prostorsko in časovno razdrobljeno skupinsko delo [Pidd, 1998, str. 214]. Dokumenti v elektronski obliki fizično ne krožijo (ves čas se nahajajo v centralni bazi podatkov), ampak se prenašajo obvestila, spreminjajo se avtorizacije uporabnikov in povezave v različne delovne mape uporabnikov.

Prehod na elektronski način poslovanja v podjetju zahteva prenovu procesa, ki izkorišča nove tehnične zmožnosti. Oblikovanje novih delovnih procesov razdelimo v naslednje korake [Duhovnik, 2000, str. 3.15]:

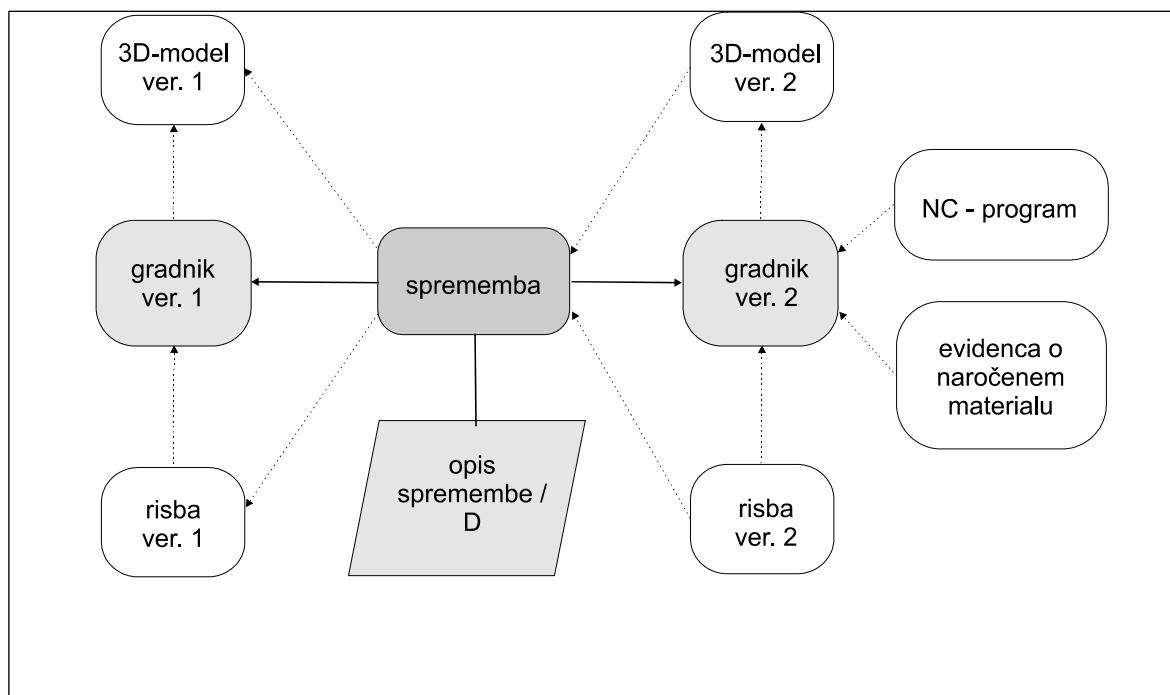
- Modeliranje informacijskih tokov (zaporedje aktivnosti, količine podatkov, pogostost izvajanja posameznih operacij...).
- Določitev pravic dostopa uporabnikov in odgovornih oseb za potrjevanje.
- Študija možnosti prenosa podatkov med dokumenti, podatkovnimi bazami, tako da imamo podatke shranjene na enem mestu.
- Postavitev potrebnih mrežnih povezav in programske opreme za pretok podatkov.

V nadaljevanju je potrebno izdelane modele postaviti v aktivno uporabo. Učinkovitost je odvisna od tega, kako postavljena konfiguracija ustreza dejanskim razmeram. Spremenjen način dela omogoča poenostavitve, ukinitve nekaterih dokumentov in izboljššan pregled med nastajanjem dokumentov.

Obvladovanje sprememb

PDM sistemi ponujajo za obravnavo sprememb predloge postopkov, ki se v podrobnostih prilagodijo potrebam konkretnega podjetja. V času razvoja ne kreiramo novih verzij, ampak dopolnjujemo nastajajočo dokumentacijo. Vse dokumente, ki so bili v nekem trenutku veljavni, hranimo in v primeru spremembe kreiramo novo verzijo dokumenta oziroma elementa.

SLIKA 10: Sprememba povezuje med seboj nove in stare verzije gradnikov in pripadajočih dokumentov.



Ob kreiranju nove verzije moramo obvezno podati vzrok nastanka nove verzije. Po potrebi kreiramo tudi nove verzije objektov, ki so pripeti na predmet spremembe. Kreiranje novih verzij omogoča, da staro stanje ostane dokumentirano. Novim verzijam dodelimo začetni status, ki omogoča spreminjanje in dopolnjevanje. Novo nastale verzije dokumentov ali elementov (izdelkov) potrebujejo ponovno potrjevanje po določenem postopku. Šele ko so vsi dokumenti potrjeni, lahko zaključimo spremembo [Duhovnik, 2000, str. 3.18].

PDM sistemi imajo različne možnosti prikaza gradnikov in dokumentov. Mogoče je nastaviti, da se v strukturi vidi le zadnje verzije. Preko povezav med gradniki in dokumenti je mogoče dostopati do starih verzij.

Projektno vodenje

Večina podjetij je tesno povezanih z vodenjem projektov. Projektno vodenje temelji na razčlenitvi projekta na posamezne aktivnosti. Med seboj neodvisne aktivnosti lahko izvajamo vzporedno. Planiranje zajema razporejanje ljudi in sredstev po aktivnostih, prepoznavanje kritičnih poti ipd. Z razno programsko opremo lahko pregledno spremljamo potek dela, imamo različne možnosti izpisov in pomoč pri vrednotenju stroškov. Opozoriti velja, da so omenjeni programi praviloma le

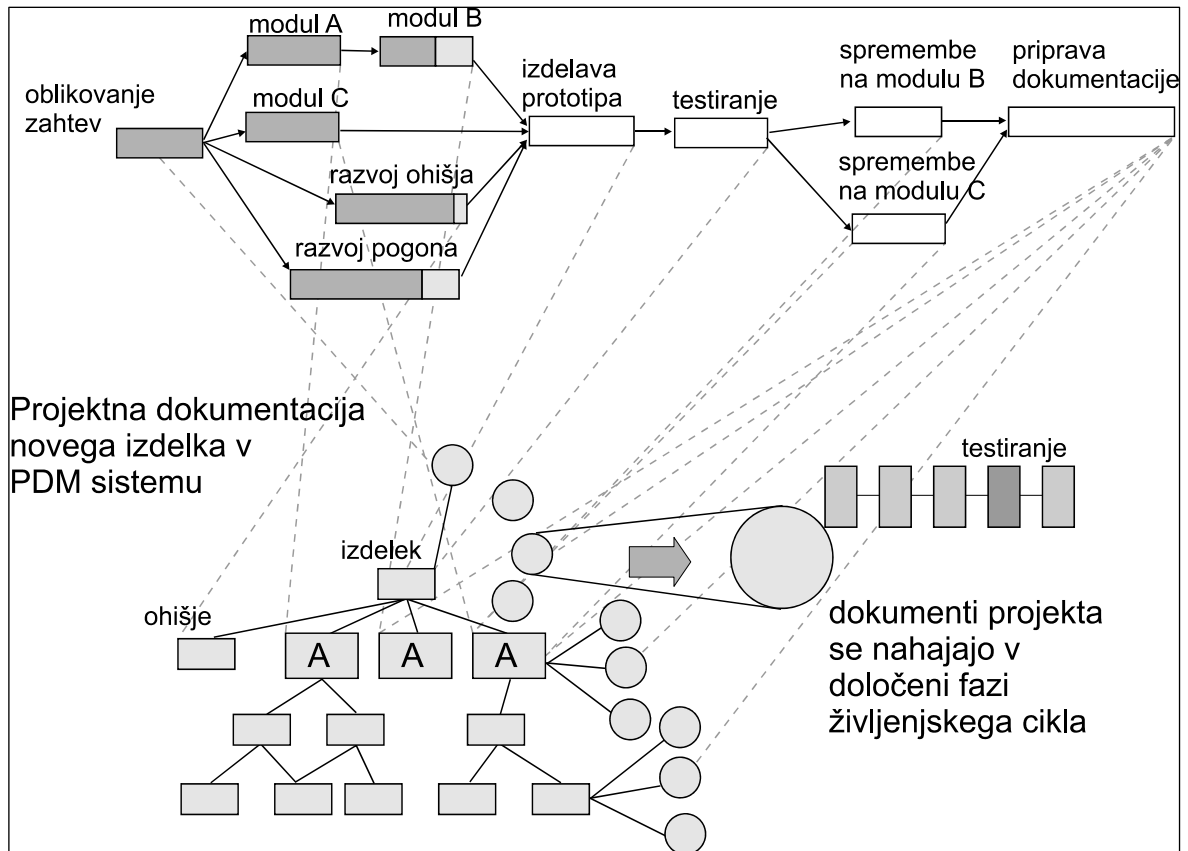
nekakšen povzetek izvajanja projekta. To pomeni, da uporabniki ročno vnašajo podatke o poteku dela, medtem ko so podatki o projektu razpršeni na različnih medijih [Stark, 1992, str. 119].

Velika prednost PDM sistemov je, da omogočajo pregled in kontroliran dostop do vseh podatkov in dokumentov, ki so vezani na projekt. Tako lahko projektna skupina, poleg vpogleda v spremljanje aktivnosti, dostopa direktno tudi do vse projektne dokumentacije in preveri, kako daleč je delo in kaj povzroča morebitne zastoje. Projektni skupini so podatki dostopni povsod, kjer je postavljen odjemalec PDM sistema. PDM sistem se uporablja kot komunikacijsko sredstvo med člani projektna skupine. Pri tem se lahko uporabnik sklicuje na vse dokumente ter enostavno in zanesljivo dokumentira vse pripombe [Spinner, 1997, str. 74].

PDM sistemi so zelo primerni za projekte, kjer je cilj izdelava dokumentacije. Učinkovito se uporablja tudi pri razvoju programske opreme [Graham, 1997, str. 205]. V PDM sistemu je napredovanje dela razvidno iz tega, v kateri fazi življenjskega cikla se izdelek ali dokument nahaja. Dobro integriran sistem za vodenje projektov zajema podatke iz PDM sistema in prikazuje napredovanje v npr. mrežnem ali Gantovem diagramu [Ward, 1995, str. 60]. PDM sistemi v večini nimajo svojega modula za vodenje projektov, ponujajo pa integracijo z najbolj razširjenimi programi za projektno vodenje.

PDM sistemi s svojo arhitekturo dajejo pogoje za učinkovito projektno vodenje. V osnovi so PDM sistemi prilagojeni delu v računalniškem omrežju in več uporabnikom. Z razširjeno uporabo WEB PDM odjemalcev in povečanih kapacitet prenosa podatkov, se odpre možnost nastanka virtualnih projektnih skupin [Turban, 1999, str. 5]. PDM sistemi so dobro orodje za arhiviranje projektnega dela. Shranjeni so vsi dokumenti in tudi časovni potek z datumi pregledov in overovitvami posameznih faz. Shranjene so vse pripombe, utemeljitve sprememb in načrtovanih stroškov ipd. Vse to omogoča temeljito izdelavo analize poteka izvajanja projekta.

SLIKA 11: Povezava med mrežnim planom razvoja izdelka in projektno dokumentacijo.



Arhiviranje elektronskih dokumentov

Z vedno večjim kopičenjem dokumentacije se pojavi potreba po elektronskem arhiviranju dokumentov. Pogostost dostopa določa medij, na katerem so podatki spravljani. Npr. če izvedene projekte pogosto uporabljamo pri pripravi novih ponudb, morajo biti hranjeni na trdem disku. Zgoščenke v omari so primerne za manjše skupine ljudi, tako da lahko zaposleni do njih sami dostopajo [Schichtel, 2002, str. 222].

Zahteve pri postavitvi arhiva:

- Možnost uporabe in upravljanja arhiva iz različnih lokacij.
- Predvidena rast količine podatkov naj ne bo ovira.
- Podatki naj bodo shranjeni neodvisno od strojne in programske opreme.
- Sistem za arhiviranje mora biti stabilen.
- Čas hranjenja podatkov za običajne izdelke je 10 let, pri nekaterih izdelkih celo 30 do 60 let.

3. UML

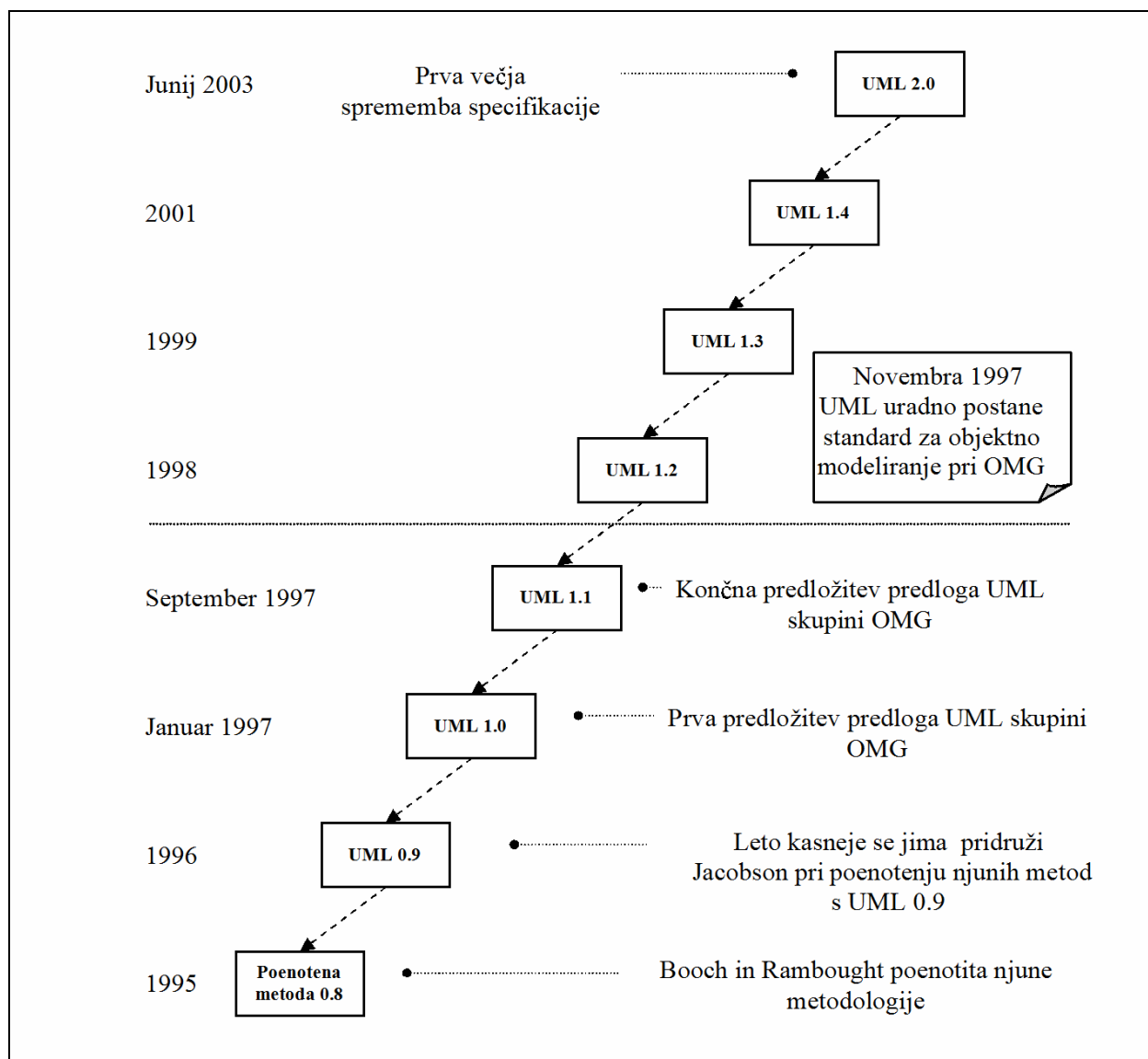
Poenoten jezik za modeliranje UML je grafični jezik za vizualizacijo, specifikacijo, načrtovanje in dokumentacijo programske opreme. UML ponuja vzorce za načrtovanje sistema, vključujoč konceptualne stvari kot so poslovni procesi in funkcije sistema, kakor tudi bolj konkretne stvari, kot so vgrajeni stavki programskega jezika, sheme baze podatkov in ponovno uporabne programske komponente.

UML predstavlja zbir najboljših postopkov v praktičnem objektno orientiranem modeliranju. Je rezultat večletnega dela, katerega cilj je bil poenotenje metod, ki so najpogostejše v uporabi pri razvoju industrijskih rešitev. Največji truda pa je bilo vloženo v poenostavitve in povečanje uporabnosti [UML 1.4, 2001, str. 19]. UML ni metodologija za razvoj informacijskega sistema, ampak jezik, skupek tehnik, ki jih lahko uporabljamo skupaj s poenotenim procesom ali katero drugo metodologijo.

3.1 ZGODOVINA

V relativno kratkem časovnem obdobju je poenoten jezik za modeliranje postal glavni modelni jezik v računalniški industriji. Poenoten jezik za modeliranje je nastal z združitvijo več objektno orientiranih metodologij. V začetku 90-tih let število objektno orientiranih metod naraste do števila 50. Številne izmed teh metod težko najdejo ustrezen modelni jezik, ki bi ustrezal njihovim specifičnim zahtevam. Med temi novimi metodami so izstopale Boochova metoda, Jacobsonova metoda OOSE (Object-Oriented Software Engineering) in Rumbaughtova metoda OMT (Object Modeling Technique). Boochova metoda je bila učinkovita v fazi načrtovanja in konstrukcije, OOSE je omogočala odlično podporo za diagrame razredov, analizo in visokonivojsko načrtovanje, OMT pa je bila najuporabnejša pri analizi in pri sistemih z veliko podatki [Booch, 1998, str. 15]. V sredini 90-tih let se avtorji vseh treh metod Boochove, OOSE in OMT odločijo za snovanje novega modelnega jezika, ki bi bil podprt v vseh treh metodah - poenotenega jezika za modeliranje [Booch, 1998, str. 15].

Slika 12: Zgodovinski razvoj specifikacije poenotenega modelnega jezika UML



Vir: UML 2001 A Standardization Odyssey, 1999

3.2 NAMEN MODELIRANJA Z JEZIKOM UML

Modele gradimo zaradi boljšega razumevanja delovanja sistema, ki ga načrtujemo [Booch, 1999, str. 6]. Namen modeliranja je doseči:

1. Boljšo vizualizacijo delovanja delujočega sistema ali sistema v razvoju. Konceptualni model omogoča lažje razumevanje sistema tako za tistega, ki ta sistem razvija, kakor tudi za nekoga drugega, ki poizkuša delovanje

le-tega razumeti. Model ne omogoča samo lažjega razumevanja sistema kot takega, ampak tudi razumevanje koncepta njegovega načrtovanja.

2. Prikaz strukture in obnašanja sistema. Zgrajen model mora biti natančna kopija modeliranega sistema in mora biti jasno definiran.
3. Izdelava šablone, ki nas vodi pri načrtovanju sistema. UML modelni jezik omogoča generacijo kode v številnih programskih jezikih. Mogoča je pretvorba modela v programske jezike kot so C, C++, Java, Visual Basic ali pa celo v relacijsko bazo podatkov.
4. Dokumentiranje procesa načrtovanja. UML omogoča dokumentacijo celotne arhitekture sistema, pisanje zahtev in testnih procedur. UML pa vsebuje tudi jezik za modeliranje aktivnosti pri načrtovanju projekta in upravljanju z izdanimi verzijami sistema.

3.3 KONCEPTUALNI MODEL

Za razumevanje UML-ja je potrebno poznati pomen treh osnovnih gradnikov:

1. elementov,
2. povezav,
3. diagramov.

ELEMENTI

1. strukturni elementi,
2. elementi obnašanja,
3. elementi združevanja,
4. elementi opomb.

STRUKTURNI ELEMENTI

Največkrat so to statični deli modela, ki predstavljajo element shematično ali fizično:

1. razredi,
2. vmesniki,
3. sodelovanja,
4. primeri uporabe,
5. komponente,
6. vozlišča.

Razred je opis nabora objektov, ki delijo enake attribute, operacije, povezave in semantiko.

Vmesnik definira povezavo med specifikacijo, ki nam pove, kaj abstrakcija počne in med implementacijo, ki pove, kako to počne. Vmesnik je zbirka operacij, ki opisujejo, kako razred ali komponenta deluje. Lahko prikazuje celotno ali pa le delno obnašanje razreda ali komponente. Največkrat so vmesniki povezani z razredom ali komponento, ki realizira vmesnik. Dobro strukturiran vmesnik omogoča jasno delitev med pogledom od zunaj in notranjim delovanjem abstrakcije, s tem omogoča razumevanje delovanja brez detajlnega poznavanja delovanja abstrakcije.

Sodelovanje je skupina razredov, vmesnikov in drugih elementov, ki delujejo skupaj v kooperativnem obnašanju, katerega rezultat je izboljšano delovanje sistema. Zapis v UML prikazuje sodelovanje navzven kot en sam kos, vendar pa sodelovanje v sebi skriva različne diagrame, največkrat so to razredni diagrami (strukturni del sodelovanja) in diagram sodelovanja (vedenjski del sodelovanja).

Primer uporabe je opis niza zaporednih akcij, ki jih izvaja sistem z namenom, da določenemu akterju prinese rezultat, ki ga lahko opazujemo.

Aktivni razred je razred, katerega objektom pripadajo eden ali več programskih procesov, ali programskih niti in lahko sam nadzoruje svojo aktivnost. Aktivni razred je razred, katerega instance so aktivni objekti.

Komponenta je fizični in zamenljivi del sistema, ki omogoča izvedbo niza vmesnikov. Komponente uporabljamo za modeliranje fizičnih stvari, ki jih vključimo s pomočjo izvršljivih programov, knjižnic, tabel, datotek in dokumentov. Ponavadi komponenta vsebuje fizični paket logičnih elementov, kot so razredi, vmesniki in sodelovanja.

Vozlišče je fizični element, ki obstaja v času izvajanja programa in predstavlja programski vir, z lastno rezervacijo pomnilnika in sposobnostjo procesorskega dostopa. Vozlišča se uporabljajo za modeliranje topologije strojne opreme, na kateri teče nas sistem. Izraz vozlišče predstavlja procesor ali napravo, na kateri se komponente uporabljajo.

ELEMENTI OBNAŠANJA

Elementi obnašanja so dinamičen del UML modela in predstavljajo časovno in krajevno obnašanje modela.

Ločimo dva tipa elementov obnašanja:

- interakcije,
- diagram stanj.

Interakcija je obnašanje, ki vključuje nabor sporočil, ki se izmenjujejo med objekti v okviru določenega konteksta za doseganje določenega cilja. Z uporabo interakcije modeliramo dinamični aspekt sodelovanja, predstavimo funkcijo delovanja skupine objektov, ki je z vzajemnim delovanjem boljše kot delovanje vsakega objekta posebej. Funkcije delovanja pa so predstavitev prototipov instanc razredov, vmesnikov, komponent, vozlišč in primerov uporabe. Vsako interakcijo lahko modeliramo s poudarkom na časovnem poteku sporočil ali pa s poudarkom na zaporedju prehajanja sporočil.

Diagram stanj je obnašanje, ki opisuje zaporedje stanj objekta ali interakcij med objekti, kot odgovor na dogodke v sistemu. Z njim lahko zelo dobro modeliramo življenjsko dobo objekta, s čimer zelo dobro vidimo, kaj se v sistemu dogaja. Diagram stanj vključuje številne elemente stanja, prehode med posameznimi objekti, dogodki (z njimi lahko sprožimo prehode med objekti) in aktivnosti (odziv na prehod).

ELEMENTI ZDRUŽEVANJA

Element združevanja je organizacijski del UML, glavni mehanizem tega procesa pa imenujemo paket, ki omogoča združevanje elementov. Za razliko od komponent, ki obstajajo v času izvajanja, je paket konceptualnega značaja. Paket nam služi za združevanje modelnih elementov na nivoju arhitekturnega načrtovanja v večje skupine, s katerimi je lažje upravljati.

ELEMENTI OPOMB

Elemente opomb uporabljamo za opisovanje in pojasnjevanje delovanja posameznih elementov v modelu. Ta element imenujemo zapisek, ki ga lahko pripnemo na element ali na skupino elementov.

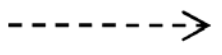
POVEZAVE

Povezava je spoj dveh elementov. V UML se pojavljajo štiri vrste povezav:

1. odvisnosti,
2. asociacije,
3. generalizacije,
4. realizacije.

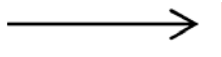
Odvisnost je semantična povezava dveh elementov, pri kateri sprememba neodvisnega elementa lahko povzroči spremembo odvisnega elementa. Odvisnost se uporablja takrat, ko poizkušamo pokazati, da ena stvar uporablja drugo.

Slika 13: Odvisnost

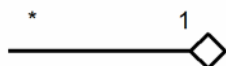


Asociacija je strukturna povezava, ki pojasnjuje nabor vezi, pri čemer so vezi povezave med objekti. Poseben primer asociacije je agregacija ali združevanje, ki predstavlja povezavo med celoto in deli te celote.

Slika 14: Asociacija



Slika 15: Agregacija



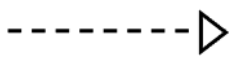
Generalizacija predstavlja povezavo specializacije in generalizacije, v kateri so objekti specializiranega elementa nadomestljivi z elementi generaliziranega elementa. Kot sem že omenil, specializiran element ima isto strukturo in obnašanje kot generaliziran element.

Slika 16: Generalizacija



Realizacija je semantična povezava med elementoma, pri kateri eden element določi dogovor, ki ga mora drugi izvršiti. Realizacija se pojavlja med vmesniki v razredih ali med komponentami, ki jih realizirajo.

Slika 17: Realizacija

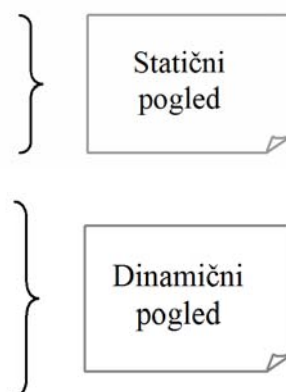


DIAGRAMI

Diagram je grafična predstavitev skupine elementov, njegov smisel pa je predstavitev sistema iz različnih vidikov. Z diagramom dosežemo ločeno obravnavo posameznih problemov v sistemu in poenostavimo modeliranje sistema. V UML ločimo strukturalne diagrame, ki predstavljajo statično strukturo sistema in diagrame, ki predstavljajo dinamično strukturo sistema.

Diagrami v UML:

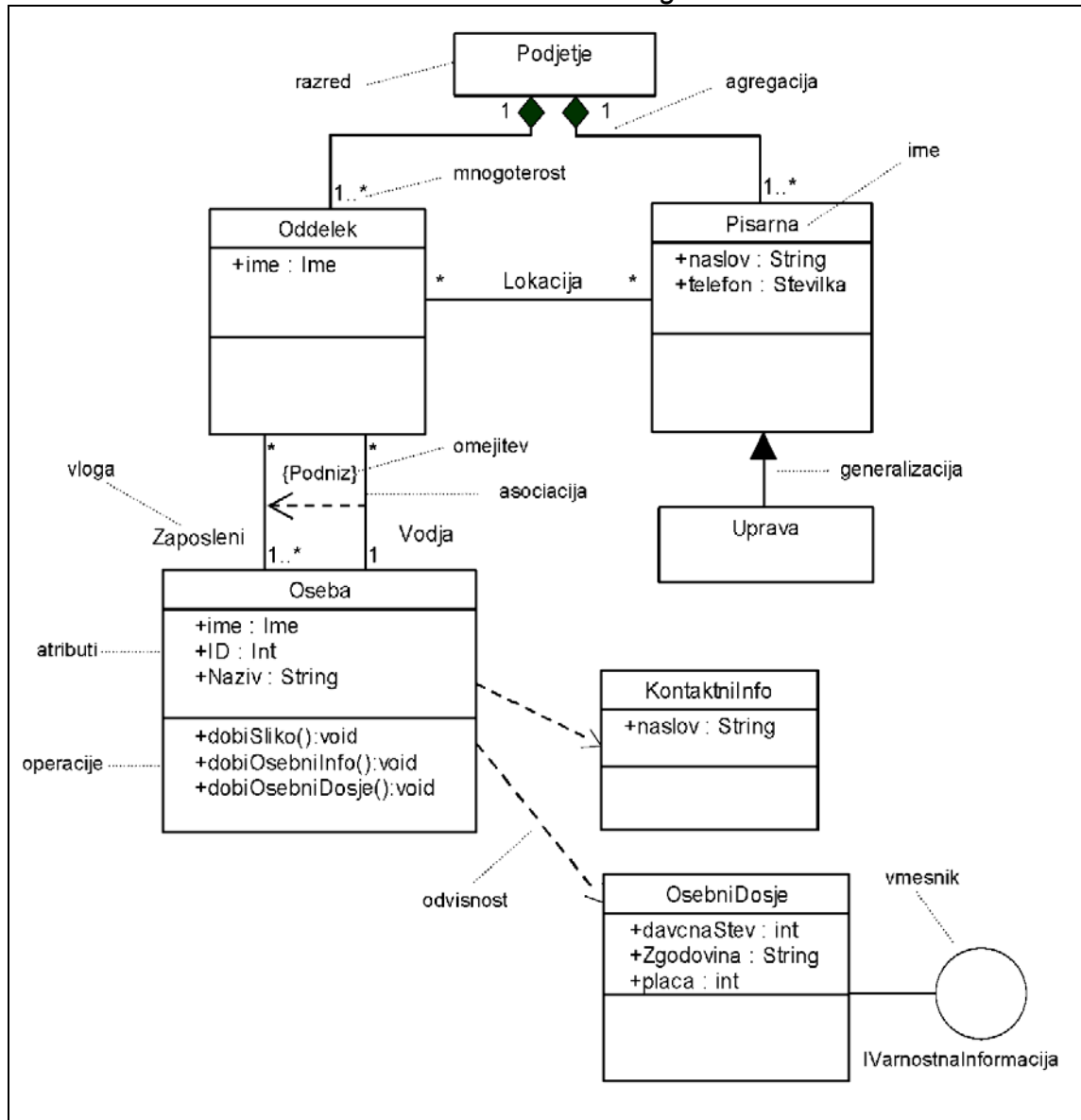
1. Razredni diagram (*ang. class diagram*).
2. Objektni diagram (*ang. object diagram*).
3. Diagram komponent (*ang. component diagram*).
4. Diagram porazdelitve (*ang. deployment diagram*).
5. Diagram primerov uporabe (*ang. use case diagram*).
6. Diagram stanj (*ang. statechart diagram*).
7. Diagram zaporedja (*ang. sequence diagram*).
8. Diagram sodelovanja (*ang. collaboration diagram*).
9. Diagram aktivnosti (*ang. activity diagram*).



Strukturni diagrami so diagrami za vizualizacijo, opisovanje, načrtovanje in dokumentiranje statičnega pogleda na sistem. Statični pogled na sistem predstavlja njegovo relativno stabilno ogrodje in zgradbo. Statični pogled opisuje funkcijo in postavitve razredov, vmesnikov, sodelovanj, komponent in vozlišč v modelu sistema.

Diagram obnašanja uporabljamo za vizualizacijo, opisovanje, načrtovanje in dokumentiranje dinamičnega pogleda na sistem. Dinamičen pogled predstavlja dele sistema, ki se spreminjajo. V sistemu dinamični pogled predstavlja tok sporočil v nekem časovnem obdobju in fizično premikanje komponent.

Slika 18: Razredni diagram



Vir: Booch, 1998, str. 106

RAZREDNI DIAGRAM

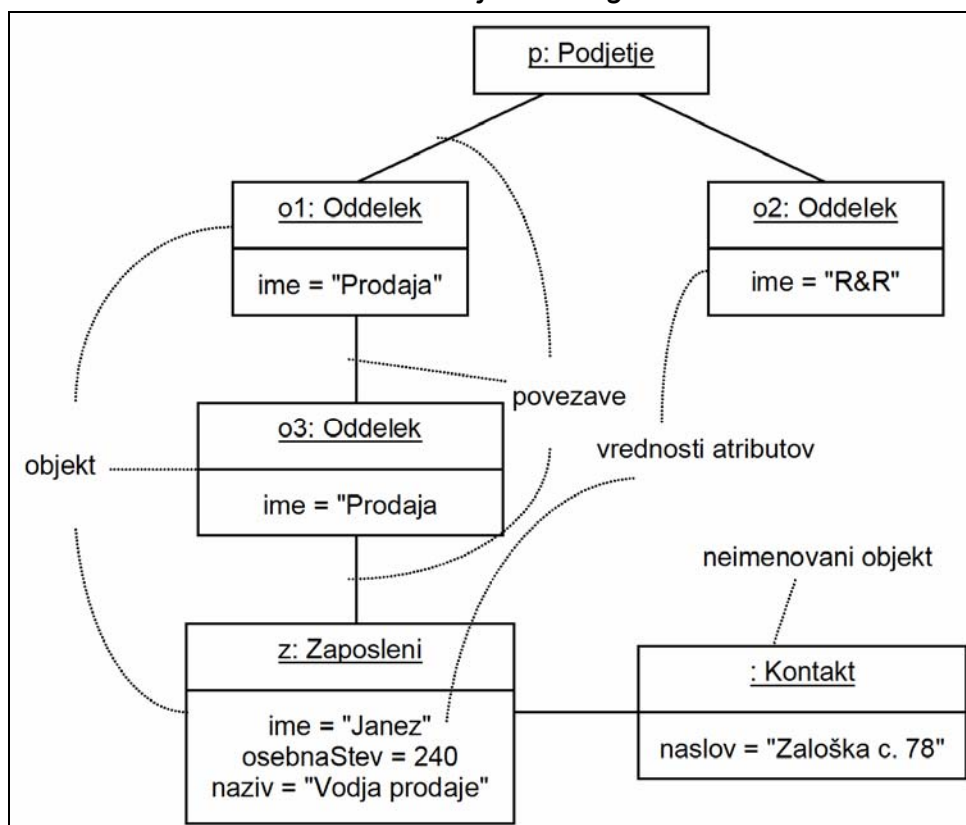
Razredni diagram, na sliki 18, prikazuje razrede, vmesnike, sodelovanja in njihove povezave. Povezave med njimi so lahko odvisnosti, generalizacije, asociacije in agregacije. Diagram lahko vsebuje tudi pakete in podsisteme, ki so namenjeni razvrstitvi elementov sistema v večje skupine. Razredni diagram je najpogostejše uporabljen diagram pri modeliranju objektno orientiranega sistema. Razredni diagrami opisujejo statični vidik načrtovanja sistema in so osnova za nekatere druge diagrame: diagram komponent in diagram porazdelitve. Niso pomembni samo za vizualizacijo, opisovanje in dokumentiranje, temveč tudi za načrtovanje izvršljivih sistemov prek vnaprejšnjega načrtovanja, generacija

programske kode neposredno iz UML modela, in vzratnega načrtovanja, prenos informacij iz programske kode v UML model.

OBJEKTNI DIAGRAM

Objektni diagram prikazuje nabor objektov in njihovih povezav. Uporabljamo jih za prikazovanje struktur podatkov in trenutnih posnetkov instanc elementov razrednega diagrama. Objektni diagrami so uporabljeni za prikaz statičnega vidika načrtovanja sistema, podobno kot razredni diagrami, z vidika realnih ali prototipnih instanc.

Slika 19: Objektni diagram

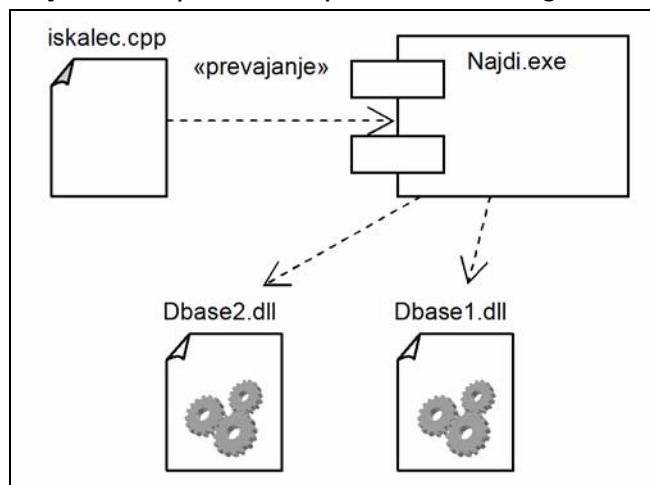


Vir: Booch, 1998, str. 196

DIAGRAM KOMPONENT

Diagram komponent na sliki 20 prikazuje organizacijo in odvisnosti med naborom komponent. Z njim je prikazan statični izvedbeni vidik sistema. Diagram komponent je tesno povezan z razrednim diagramom, saj komponenta predstavlja enega ali več razredov, vmesnikov ali sodelovanj. Uporabljamo ga za modeliranje in upravljanje z izvršljivimi elementi, knjižnicami, tabelami, datotekami in dokumenti.

Slika 20: Iskanje v bazi podatkov, prikazano z diagramom komponent



Slika 21: Iskanje v bazi podatkov preko intraneta, prikazano z diagramom porazdelitve

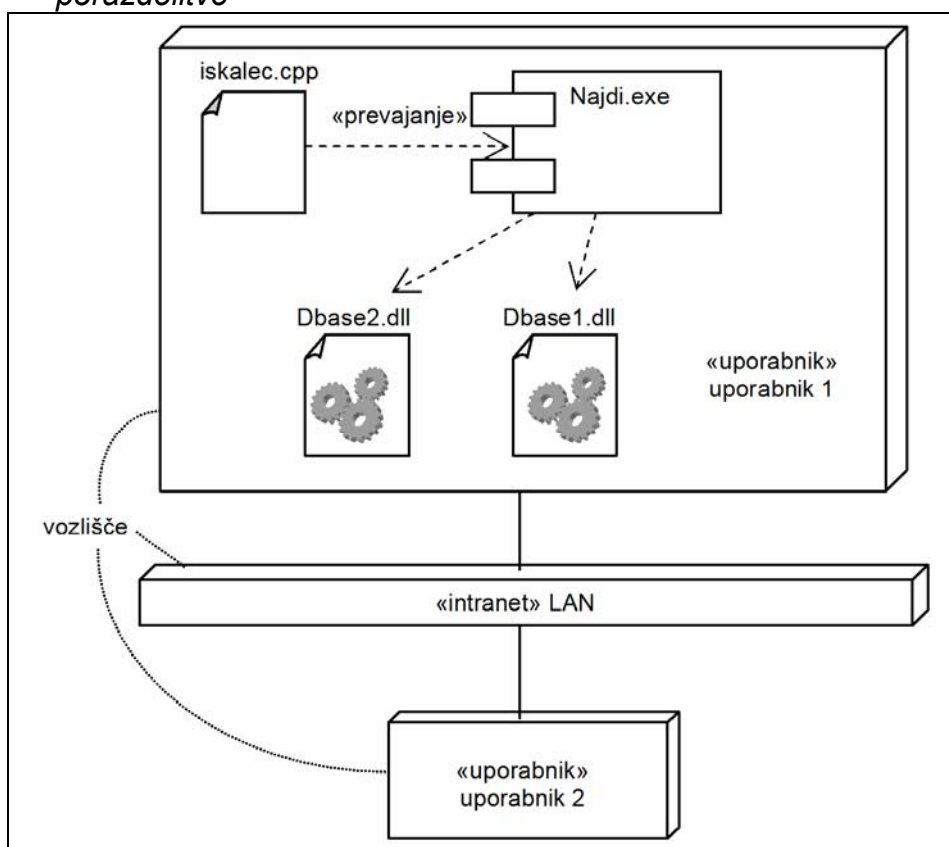


DIAGRAM PRIMEROV UPORABE

Diagram primerov uporabe prikazuje obnašanje sistema. Diagram primerov uporabe sestavljajo primeri uporabe, akterji in njihove povezave. Njihov

osnovni namen je lažje in hitrejšo zajemanje, ter predstavitev uporabnikovih zahtev. Diagrami primerov uporabe povežejo akterje -uporabnike sistema in procese, ki v sistemu tečejo. S tem omogočajo modeliranje z uporabnikovim sodelovanjem skozi celoten razvojni cikel. Služijo kot sredstvo za zajemanje in sledenje uporabnikovim zahtevam o izgledu in uporabnosti sistema.

Akterji ponazarjajo okolje. Okolje prestavlja posameznike, naprave ali drage sisteme zunaj modeliranega sistema, ki z njim komunicirajo in s tem vplivajo na njegovo delovanje. Akter tako predstavlja abstrakcijo vlog uporabnika sistema. *Primer uporabe* predstavlja zaporedje transakcij v sistemu in praviloma komunicira z akterjem. *Povezave* v diagramu primerov uporabe:

- Asociacija sproži ali zgolj sodeluje s primerom uporabe.
- Razširja: Pomeni odvisnost med primeri uporabe. Neki primer uporabe razširja drugega, kadar se le ta izvede le v določenih primerih.
- Vključuje: Kadar je obnašanje primera uporabe skupno več kot enemu primeru uporabe, tedaj rečemo, da le ta vključuje sledeče primere uporabe.
- Generalizacija: Povezava med dvema akterjema ali primeroma uporabe. Če je primer A generaliziran iz primerka B, to pomeni, da je primer B poseben primer primerka A.

Slika 22: Primer realizacije primera uporabe za bančni sistem

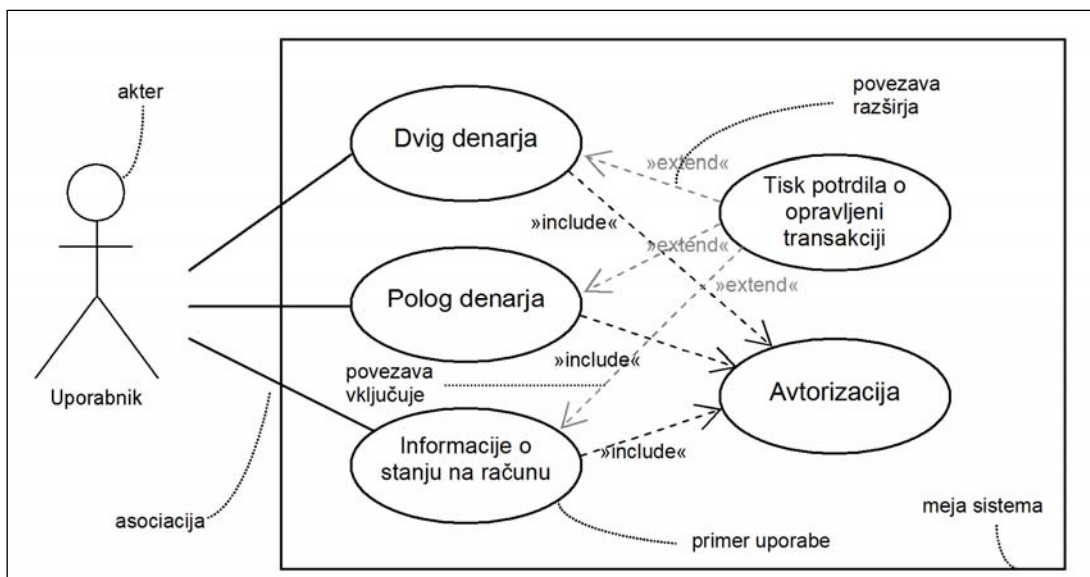


DIAGRAM STANJ

Diagrami stanj omogočajo opisovanje obnašanja sistema ali posameznega razreda. Največkrat z njim modeliramo obnašanje objektov, katerih obnašanje je odvisno od dogodkov zunaj njihovega konteksta.

Diagrami stanj prikazujejo zaporedje stanj in dogodke, ki sprožijo prehode med stanji. *Stanje* (ang. *State*) je okoliščina v življenjski dobi nekega objekta, v katerem le-ta izpolnjuje nek pogoj. Dogodek pa je pripetljaj nekega dražljaja, ki lahko sproži prehod med stanji. *Prehod* (ang.: *Transition*) je povezava med dvema stanjema.

Slika 23: Procedura za dvig denarja preko bankomata realiziran s pomočjo diagrama stanj

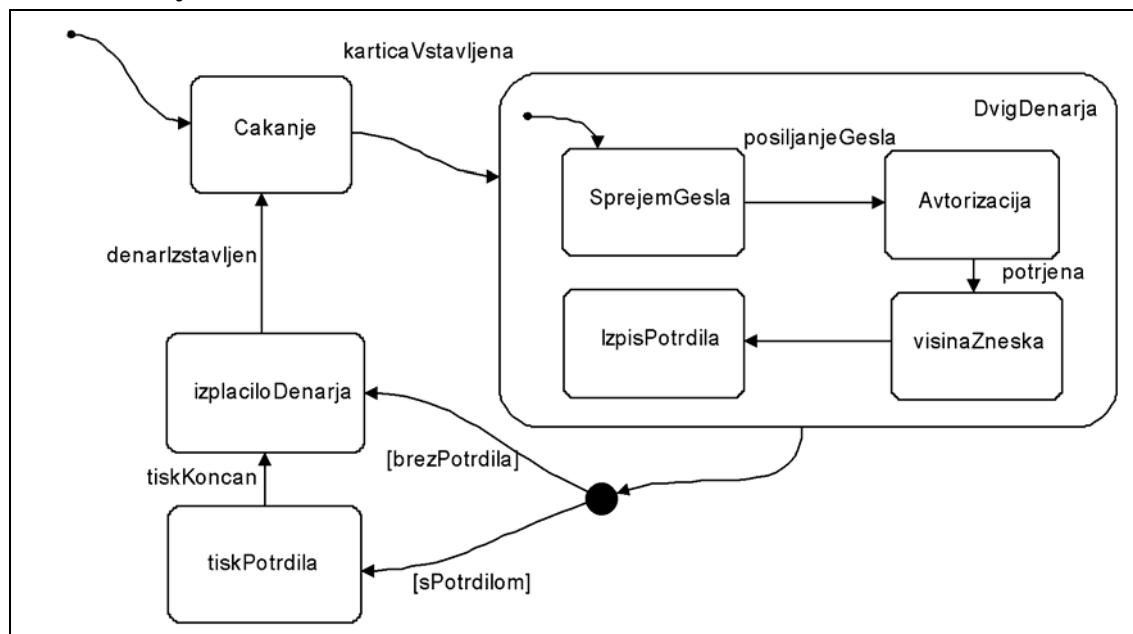


DIAGRAM ZAPOREDJA

Diagram zaporedja prikazuje časovni potek izmenjave sporočil med objekti v sistemu. Elementi diagrama zaporedja so objekti, njihove povezave in sporočila med njimi. Uporabljamo jih za ustvarjanje scenarijev v diagramu primerov uporabe. Diagram zaporedja nam pomaga razumeti povezave in medsebojne vplive med objekti s prikazom sporočil izmenjanih v nekem časovnem intervalu.

Slika 24: Procedura za dvig denarja preko bankomata z diagramom zaporedja

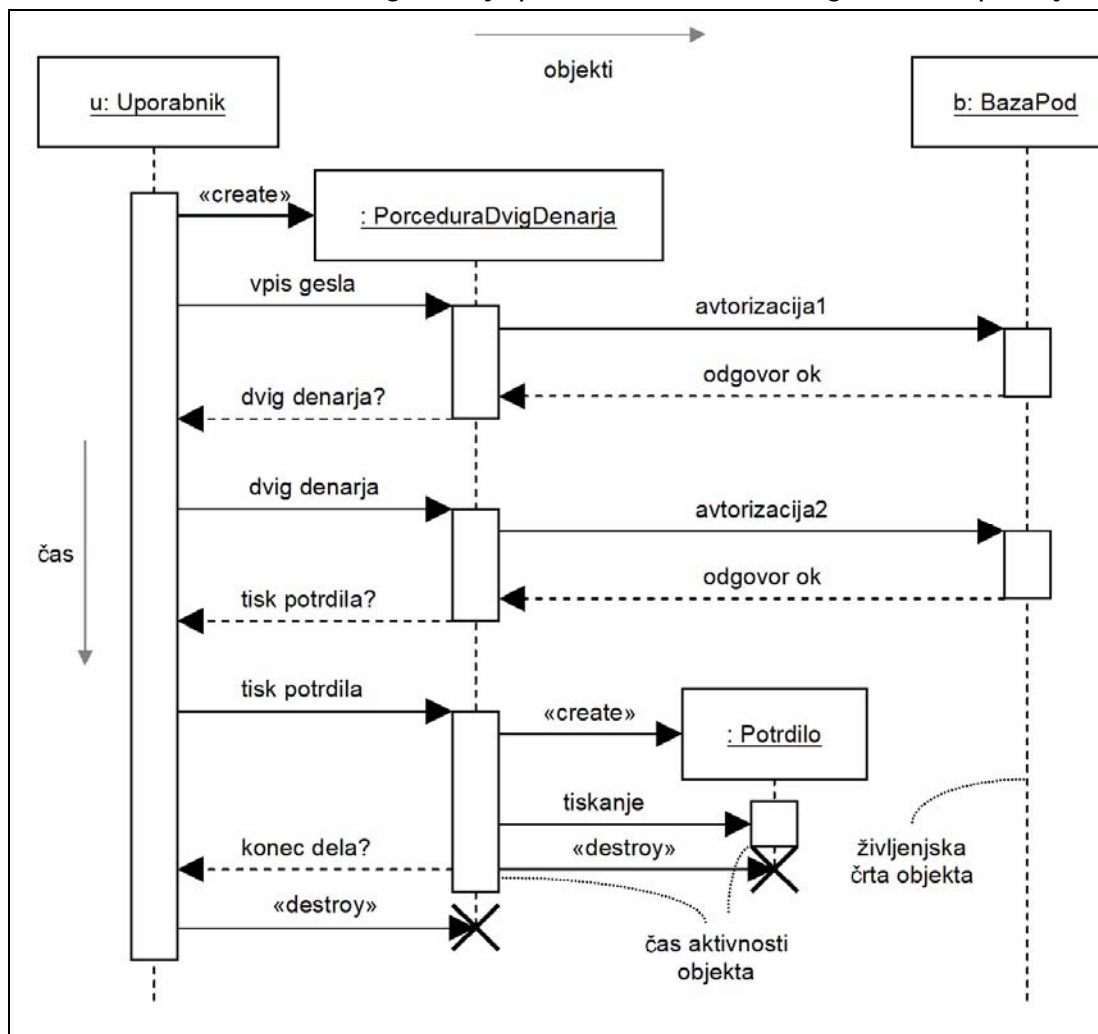


DIAGRAM SODELOVANJA

Diagram sodelovanja je zelo podoben diagramu zaporedja z bolj poudarjeno strukturno organizacijo objektov, ki pošiljajo in sprejemajo sporočila. Diagram zaporedja bolj poudari zaporedje poslanih sporočil med objekti, diagram sodelovanja pa povezave med objekti. Načrtovanje diagrama sodelovanja začnemo s postavitvijo objektov, ki sodelujejo v interakciji. Objekte nato povežemo s povezavami, katerim potem dodamo še sporočila, ki jih objekt sprejema in pošilja. Takšen diagram omogoči preglednost vseh poslanih sporočil med povezanimi objekti.

Slika 25: Procedura za dvig denarja preko bankomata z diagramom sodelovanja

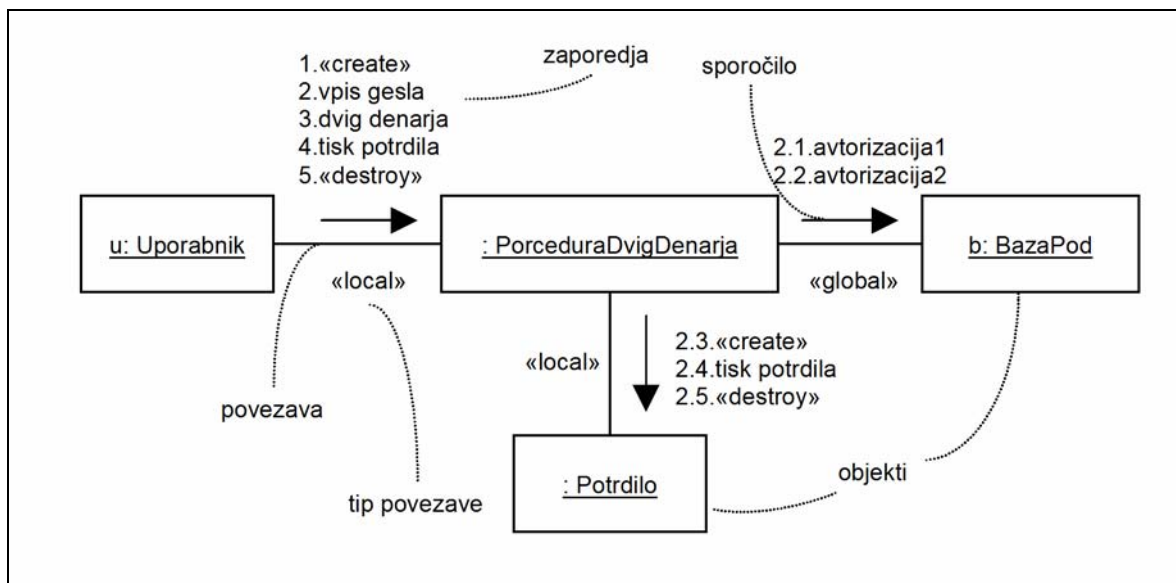


DIAGRAM AKTIVNOSTI

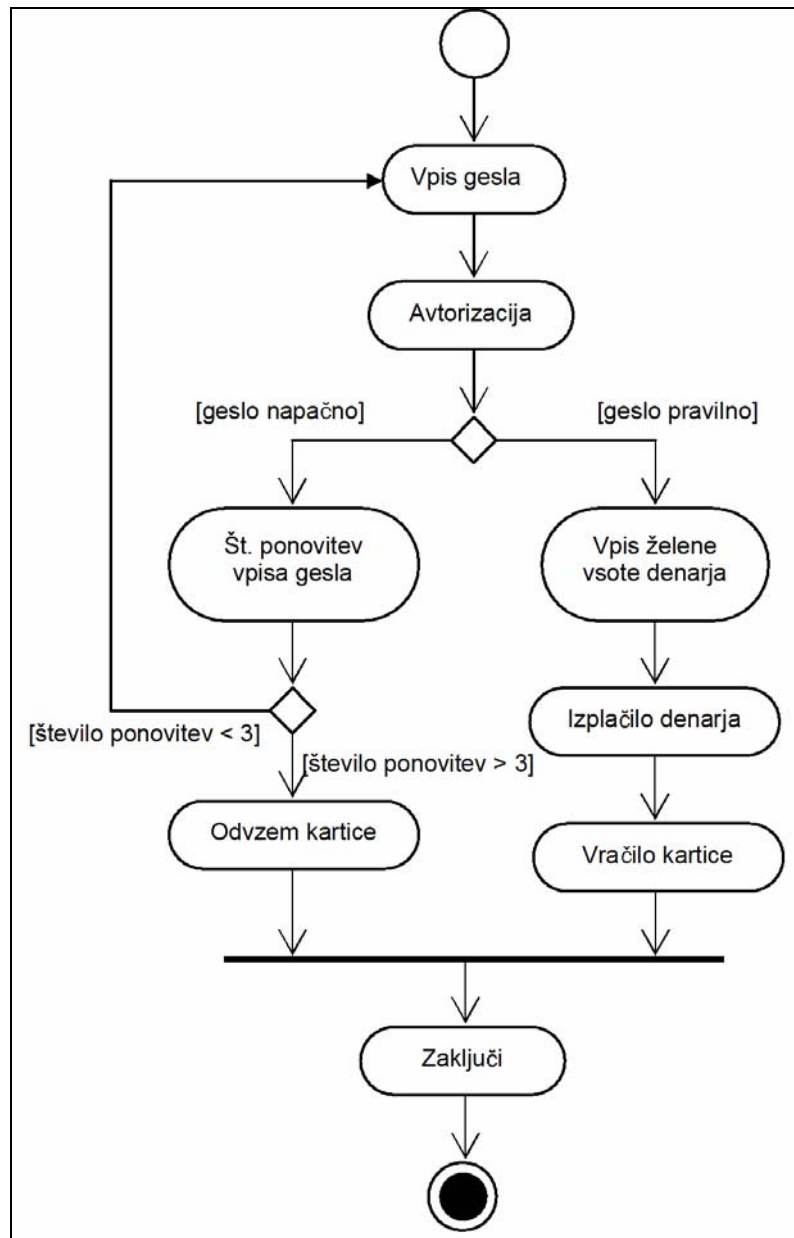
Če diagram sodelovanja prikazuje prehode iz objekta na objekt, nam diagram aktivnosti prikazuje prehode iz aktivnosti na aktivnost.

Elementi diagrama aktivnosti:

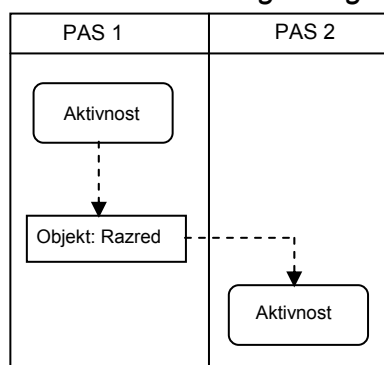
- aktivnosti,
- prehodi med aktivnostmi,
- razvejitve, sočasne razvejitve, (ang. Concurrent fork)
- sočasno stičišče, (ang. Concurrent join)
- linija toka dogodkov, (ang. Swimlane)
- objektni tok.

Rezultat aktivnost je akcija, katere posledica je prehod v drugo aktivnost, ali pa le vračanje vrednosti. Diagram omogoča navadne razvejitve ali pa sočasne razvejitve. Le-te omogočajo sočasno izvajanje dveh različnih procesov, katerih sočasno izvajanje se konča v sočasnem stičišču. Linija toka dogodkov v praksi omogoča razdeljevanje toka delovanja poslovnih procesov po posameznih poslovnih enotah ali skupinah (pasovih); iz diagrama je jasno razvidna odgovornost za posamezne aktivnosti. Če so v toku dogodkov vpleteni tudi pomembni objekti, jih lahko s pomočjo objektnega toka dodamo v diagram aktivnosti.

Slika 26: Procedura za dvig denarja preko bankomata z diagramom aktivnosti



Slika 27: Prikaz razdelitve aktivnostnega diagrama s pomočjo pasov



MODELIRANJE SISTEMA Z RAZLIČNIH VIDIKOV

Modeliranje sistema iz različnih vidikov prinese sočasno načrtovanje z večimi dimenzijami hkrati. Vendar pa je potrebno pri takšnem vidiku paziti, kajti v mislih moramo imeti celo sliko načrtovanega sistema in se ne smemo osredotočiti le na en vidik načrtovanja sistema. Potrebno je raziskati, kateri od petih vidikov najbolj prikazuje arhitekturne lastnosti modeliranega sistema.

Za modeliranje enostavnega sistema ne potrebujemo celotnega nabora diagramov, ampak je za potrebe modeliranja dovolj uporabiti le diagram primerov uporabe, razredni diagram in interakcijski diagram.

Pri kompleksnejših sistemih pa smo prisiljeni poseči po celotnem naboru diagramov. Nekateri diagrami nam postrežejo z boljšo predstavitvijo enega vidika, drugi pa so boljši za predstavitev drugega vidika [Booch, 1998, str. 99]:

- Vidik primerov uporabe: —> Diagrami primerov uporabe
 —> Diagrami aktivnosti (dinamični pogled)
- Načrtovalni vidik: —> Razredni diagrami (statični pogled)
 —> Interakcijski diagrami (dinamični pogled)
 —> Diagram stanj (dinamični pogled)
- Procesni vidik: —> Razredni diagrami (statični pogled)
 —> Interakcijski diagrami (dinamični pogled)
- Izvedbeni vidik: —> Diagram komponent
- Porazdelitveni vidik: —> Diagram porazdelitve

Sistem lahko modeliramo z različnih vidikov, poleg tega pa lahko sistem modeliramo na različnih nivojih razumljivosti. Potreba po tem nastane, če želita isti model sistema uporabljati dva različna uporabnika npr. analitik in programer. Analitik uporablja scenarije primerov uporabe, pri katerih je podrobno delovanje samega sistema zanj nepomembno in samo zamegli njegov pogled. Povsem drugače pa je s programerjem, ko potrebuje dostop do operacij in atributov. Zato lahko rečem, da uporabnika potrebujeta dva različna nivoja razumljivosti modela sistema.

4. Intervju

Poslovni proces se začne pri ključnem uporabniku, ki najprej kreira zahtevo po izvedbi prilagoditve in jo pošlje svetovalcu. Pod akcijo kreiranja zahteve spada izdelava vse dokumentacije povezane s prilagoditvijo, kreiranje artikla – prilagoditve in povezava vseh datotek - integracija s kreiranim artiklom-prilagoditvijo.

Svetovalec preveri upravičenost zahteve po izvajanju prilagoditve in po bazi podatkov poišče morebitne že izdelane prilagoditve, ki se bi jih lahko uporabilo ali predelalo. Nato dopolni zahtevek z dodatnim opisom funkcionalnosti in oceno potrebnih dni za izdelavo prilagoditve in zahtevek pošlje vodji projekta (VP) – ITS.

VP – ITS najprej preveri določila pogodbe s stranko in nato izvajanje prilagoditve direktno odobri (če je izvajanje prilagoditev že zajeto v pogodbi), ali pa pošlje zahtevek v potrditev vodji projekta – stranka. Ta ima dve možnosti. Lahko zahtevo po prilagoditvi zavrne in jo pošlje v bazo zavrnjenih prilagoditev in hkrati tudi nazaj v VP – ITS ter svetovalcu, ki pa imata še vedno možnost dokumentacijo dopolniti in ponovno poslati VP - stranka. V primeru dokončne zavrnitve mora biti iz dokumentacije razvidno, zakaj je do zavrnitve prišlo in kako se je problem rešil po drugi poti. To omogoča drugim svetovalcem kasnejši vpogled o alternativni rešitvi potencialno enakega ali podobnega problema. Druga možnost je, da vodja projekta - stranka prilagoditev odobri in jo pošlje nazaj VP – ITS, ki obvesti svetovalca in pošlje zahtevek v razvojni oddelek vodji razvijalcev.

Vodja razvijalcev razporedi nalogo za prilagoditev razvijalcu. Razvijalec prilagoditev izvede, po potrebi dopolni dokumentacijo in jo pošlje v testiranje svetovalcu, ki je prilagoditev zahteval.

Svetovalec narejeno prilagoditev preveri in jo po potrebi pošlje nazaj razvijalcu v dodelavo (seveda s pripadajočimi komentarji in dokumentacijo). Razvijalec popravi napake in jo spet pošlje nazaj. Ta krog se lahko ponavlja dokler ni prilagoditev zadovoljiva in ustrezna.

Ko se svetovalec odloči, da je prilagoditev gotova, jo pošlje še v testiranje stranki. Stranka – ključni uporabnik (KU) nato prilagoditev testira in pošilja v popravilo dokler ni zadovoljna z novimi funkcionalnostmi. Ko prilagoditev potrdi, jo svetovalec razporedi v bazo gotovih prilagoditev (v pripadajočo glavno skupino in

podskupine). Razvrstitev je pomembna zaradi kasnejšega lažjega iskanja že izvedenih prilagoditev.

Prav tako je smiselna tudi razvrstitev zavrženih zahtev po prilagoditvi. Razlog je tudi v tem primeru v kasnejšem iskanju po zahtevah, saj je lahko iz razloga zavrnitve in alternativne rešitve razvidna rešitev kasnejše težave.

5. Načrtovanje PDM sistema v podjetju s pomočjo UML metode

Glede na to, da v našem primeru uvajamo že izdelan programski paket, ki ga je potrebno prilagoditi potrebam podjetja, ne bom uporabil vseh UML diagramov, ampak samo osnovne (primere uporabe in diagram primerov uporabe, interakcijske diagrame, razredni diagram in aktivnostni diagram). Na podlagi izdelanih osnovnih diagramov bom lahko izluščil vse potrebne informacije za pripravo načrta uvajanja PDM sistema.

5.1 Primeri uporabe

S primeri uporabe prikažemo obnašanje sistema, ki ga razvijamo, pri tem pa ni potrebno podrobno opisati, kako je obnašanje izvedeno. Na osnovi prejšnjega intervjuja opišemo primere uporabe, ki opisujejo potrebne funkcionalnosti sistema. Tabela 1 in tabele 2 do 8 (priloga) prikazujejo primere uporabe kreiranje zahteve za prilagoditev, preverjanje upravičenosti prilagoditve in pošiljanje VP – ITS, potrditev izdelave prilagoditve, razporeditev izvajanja dela razvijalcu, v testiranje svetovalcu, potrditev svetovalca in v testiranje stranki, potrditev prilagoditve s strani stranke in razporeditev prilagoditve v bazo gotovih prilagoditev.

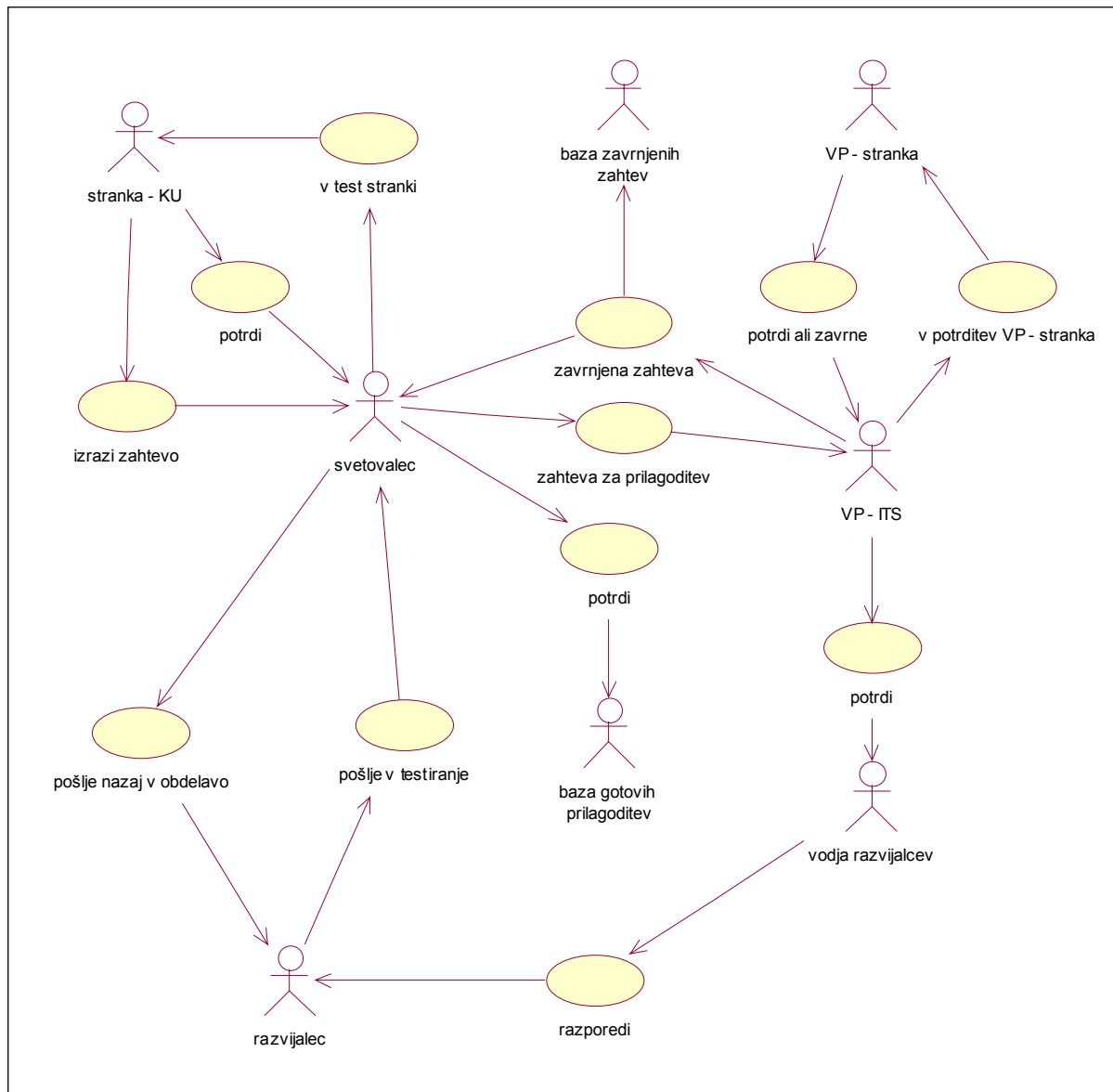
Primere uporabe predstavimo z diagramom primera uporabe, ki pokaže povezavo med aktivnimi udeleženci in primerom uporabe sistema. Diagrami vključujejo povezave med akterjem in primerom uporabe, generalizacije med akterji in generalizacije tipa razširja <<extend>> in uporablja <<uses>> med primeri uporabe.

Tabela 1: Kreiranje zahteve za prilagoditev

<u>PRIMER UPORABE: KREIRANJE ZAHTEVE ZA PRILAGODITEV</u>
<u>Pregled</u> Namen primera uporabe je kreiranje zahteve za prilagoditev
<u>Primarni akter</u> Ključni uporabnik
<u>Sekundarni akter</u> Ni
<u>Začetna točka</u> Akter ugotovi potrebo po novi funkcionalnosti
<u>Končna točka</u> Zahtevek za prilagoditev je kreiran in vnesen v sistem
<u>Izmerljiv rezultat</u> Zahtevek za prilagoditev
<u>Diagram poteka</u> Akter izrazi zahtevo po izdelavi nove funkcionalnosti – prilagoditvi in izdela dokumentacijo. Dokumentacijo skupaj s svetovalcem za področje pregledata, dopolnita in vneseta v sistem.
<u>Alternativen diagram poteka</u> Ni
<u>Razširitve primera uporabe</u> Ni
<u>Nerešeni problemi</u> Ni

Diagram 1 prikazuje diagram primerov uporabe. Naredil sem ga s pomočjo primerov uporabe, ki sem jih pripravil s pomočjo intervjuja. Diagram primerov uporabe v našem primeru prikazuje povezave med uporabniki sistema in omogoči nazoren prikaz zelenega stanja sistema po nastavitvah. To omogoči dobro osnovo za preverjanje razumevanja dejanskih potreb stranke.

Diagram 1: Diagram primera uporabe – potrditev in razvoj nove prilagoditve

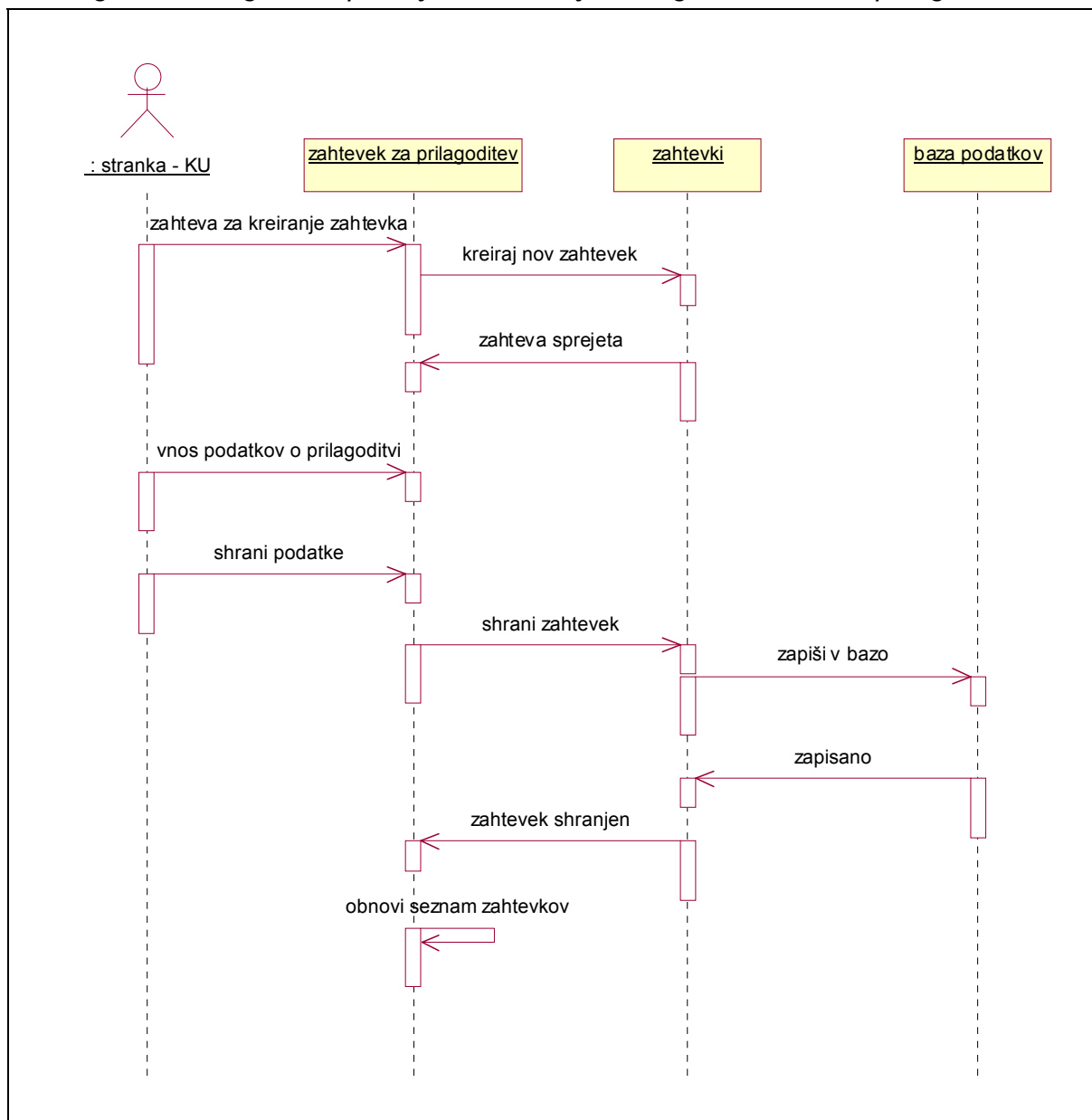


5.2 Interakcijski diagrami

Poznamo dva tipa interakcijskih diagramov:

- diagrame zaporedja
- diagrame sodelovanja

Diagram 2: Diagram zaporedja za kreiranje novega zahtevka za prilagoditev



Obe vrsti diagramov sta enakovredni saj prikazujeta isto informacijo, le v načinu predstavitve se razlikujeta. Diagrami zaporedja imajo poudarjeno časovno komponento, grafično opisujejo aktivne udeležence, objekte in sporočila med njimi. Diagrami sodelovanja pa so bolj osredotočeni na sodelovanje med objekti in akterji. Za katerega se odločimo pa je stvar naše presoje.

V nalogi sem pripravil diagrame zaporedja in diagrame sodelovanja. Naredil sem jih na osnovi diagrama primerov uporabe, ki pove osnovne značilnosti sistema. Na podlagi poznavanja funkcionalnosti programskega paketa in možnosti prilagoditve paketa sem te osnovne značilnosti razširil in jih zapisal v obliki primerni za

nastavitev sistema. Na podlagi izkušenj sem dodal možnosti povezovanja zahteve za prilagoditev v različne mape in mesta, kjer se kreirajo avtomatska elektronska obvestila. Diagrami služijo kot nazoren prikaz delovanja sistema in lahko pomagajo pri razumevanju problematike in možnosti sistema v fazi svetovanja stranki.

Diagram2a: Diagram sodelovanja za kreiranje novega zahtevka

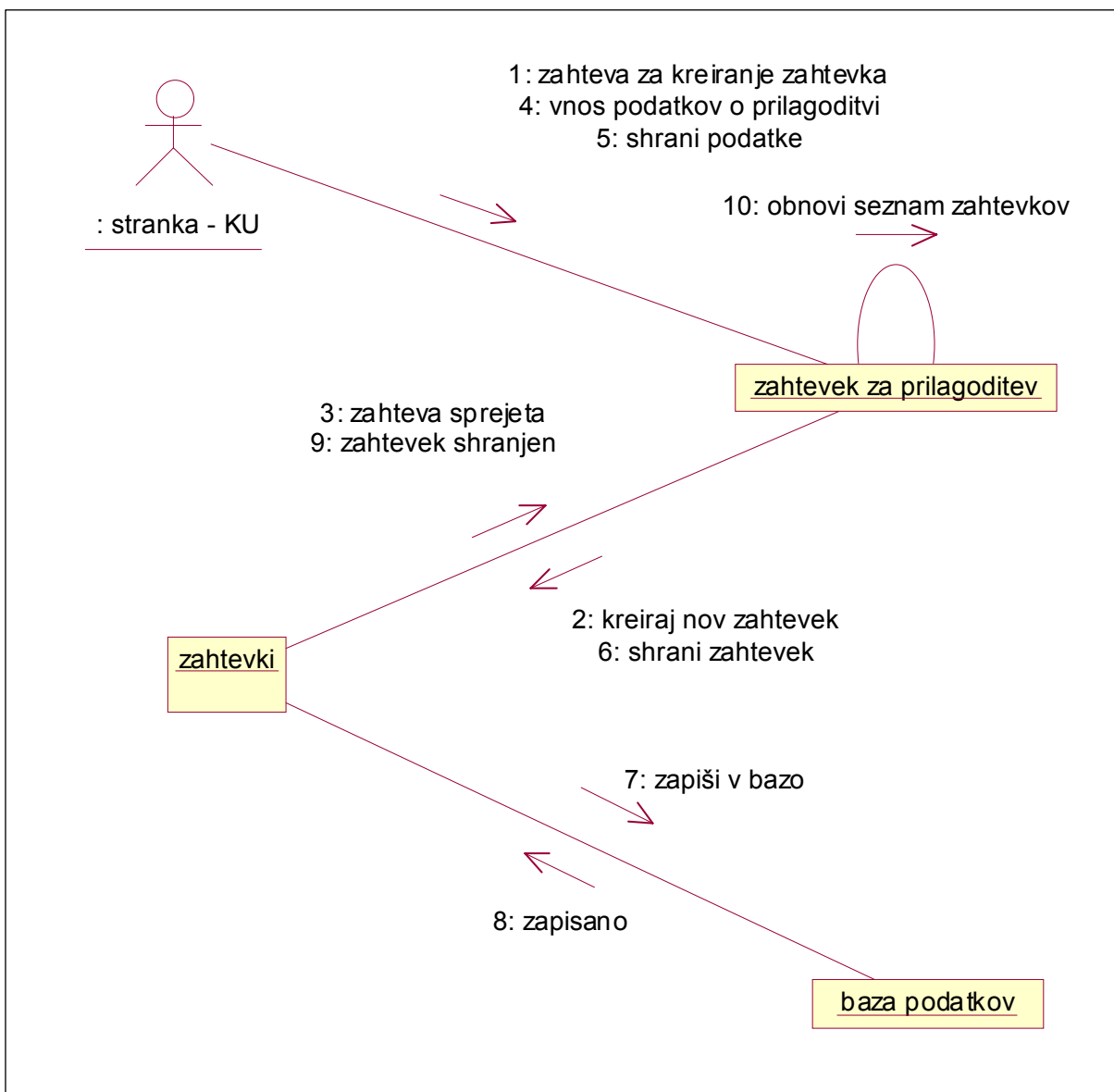
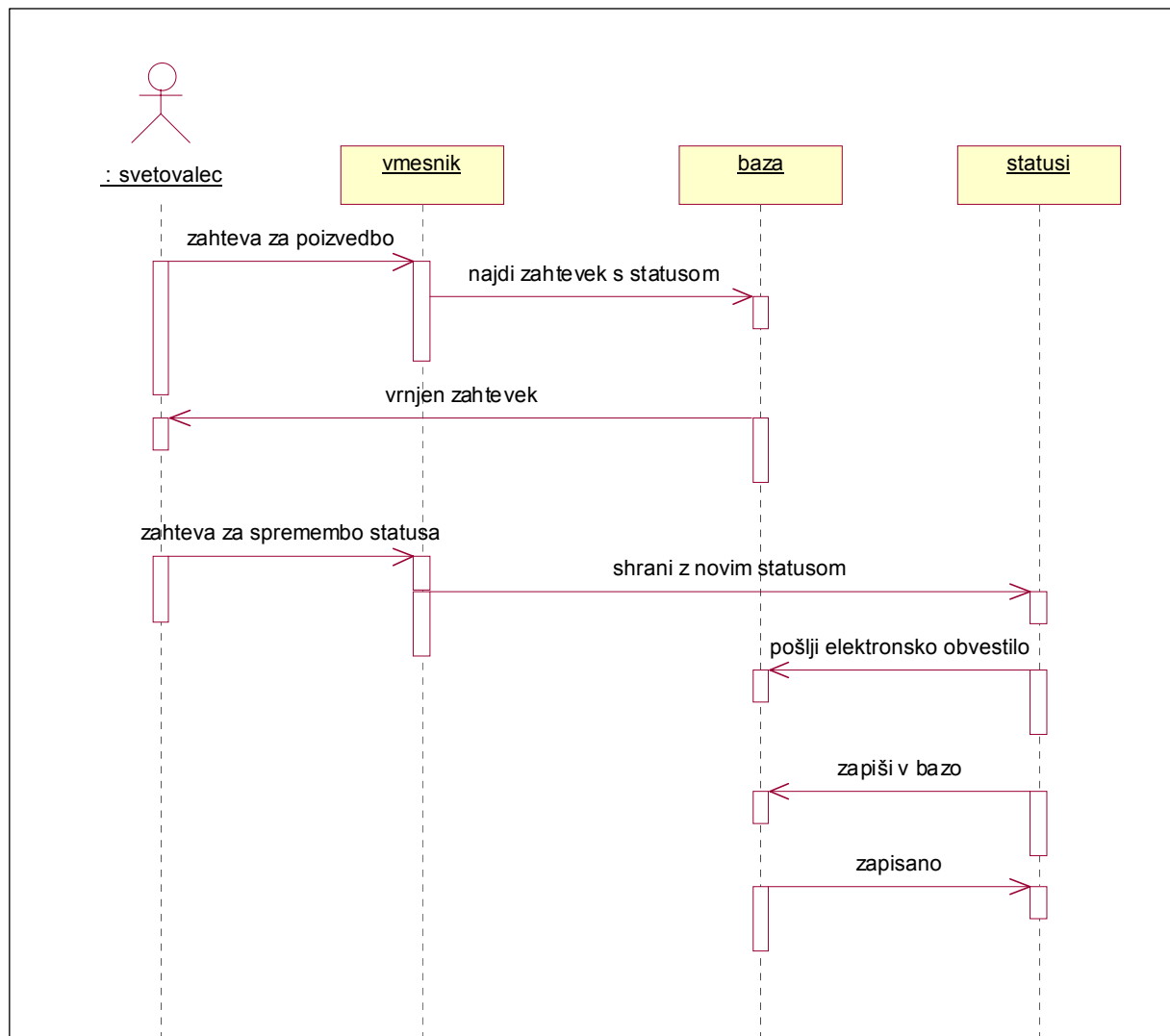


Diagrama 2 in 2a prikazujeta sorazmerno enostaven diagram za kreiranje novega zahtevka. Iz diagrama lahko razberemo na kakšen način se kreirajo novi zahtevki za prilagoditev, torej kako sodelujemo s sistemom in po kakšnem zaporedju. Diagrama 3 in 3a prikazujeta pošiljanje zahtevka VP – ITS. Diagrama 4 in 4a (priloga) prikazujeta potrjevanje zahtevka. Ostali diagrami (priloga) prikazujejo

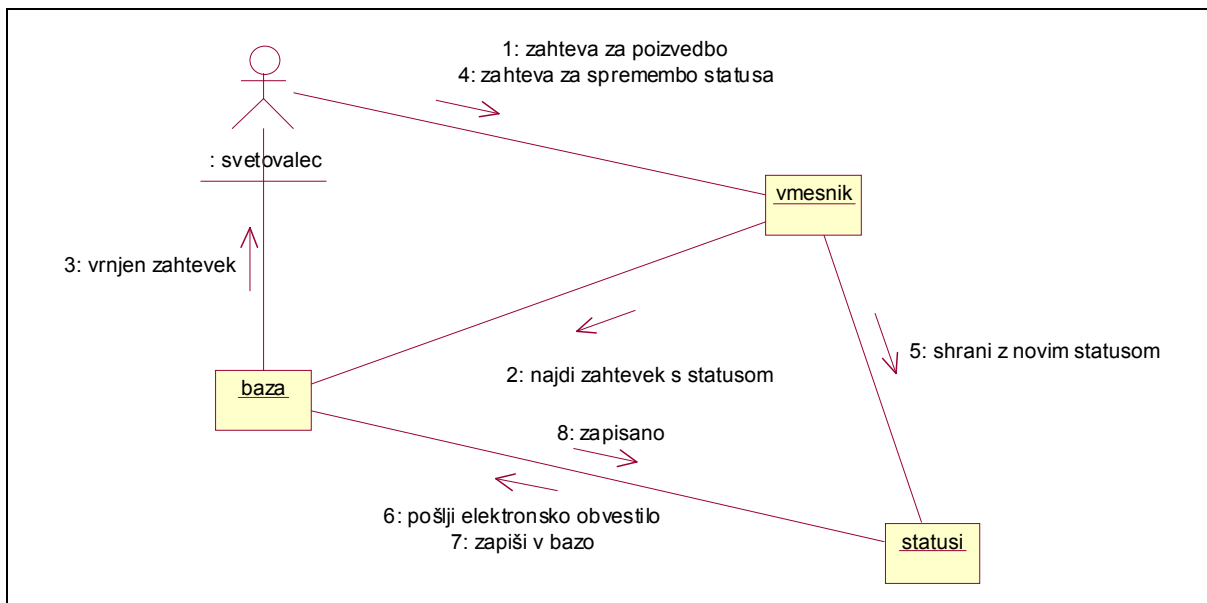
nadaljnji tok dokumenta v razvoju in so si podobni, saj vse spremembe potekajo na osnovi sprememb statusa dokumenta, ki sproži akcijo avtomatskega pošiljanja elektronskega obvestila in vezavo dokumenta v druge mape.

Diagram 3: Diagram zaporedja za pošiljanje zahtevka VP – ITS (ustrezna sprememba statusa)



S pomočjo interakcijskih diagram sem lahko podrobno opisal delovanje sistema, kar pa je v primeru uvajanja obstoječega programskega paketa celo prepodrobno, saj bi v primeru kompleksnejših zahtev porabili ogromno časa za preveč podroben opis delovanja sistema, ki je že narejen.

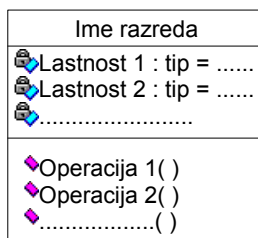
Diagram 3a: Diagram sodelovanja za pošiljanje zahtevka VP – ITS (ustrezna sprememba statusa)



5.3 Razredni diagrami

Na osnovi intervjujev in primerov uporabe identificiramo razrede objektov. Razred je opis za množico objektov, ki imajo podobno strukturo, obnašanje in povezave. Delamo po naslednjem postopku: iz intervjujev in analize primerov uporabe poiščemo vse kandidate za razrede. Kandidati so samostalniki, ki smo jih zapisali pri intervjuju. Te samostalnike obravnavamo kot razrede. Izločimo redundantne razrede, ki opisujejo isto informacijo, ki nimajo povezave s problemom, nato pa tiste, ki niso povsem jasno definirani. Nato dodamo še razrede, ki se nam na osnovi izkustvenega znanja zdijo smiselni pri reševanju problema.

Razred predstavimo na naslednji način:



V našem primeru dobimo naslednjo paleto razredov:

- svetovalec,
- VP - ITS,
- vodja razvijalcev,
- razvijalec,
- KU,
- VP – stranka,
- Zahteva za prilagoditev.

Identifikacija povezav - asociacij med objekti

Pridobimo jih na osnovi primerov uporabe in izkušenj. Povezava pomeni odvisnost med dvema razredoma. Obsegajo neposredne povezave, komunikacije, lastništvo fizične lokacije, izpolnitev določenih pogojev. Med razredi imamo lahko naslednje povezave:

Slika 28: povezave med razredi

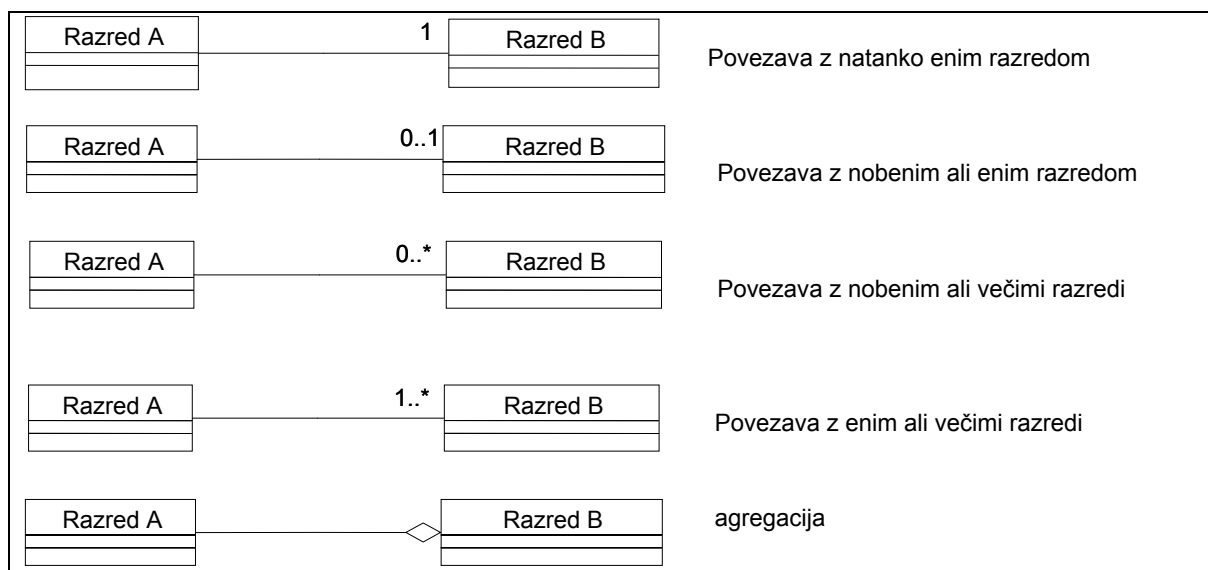
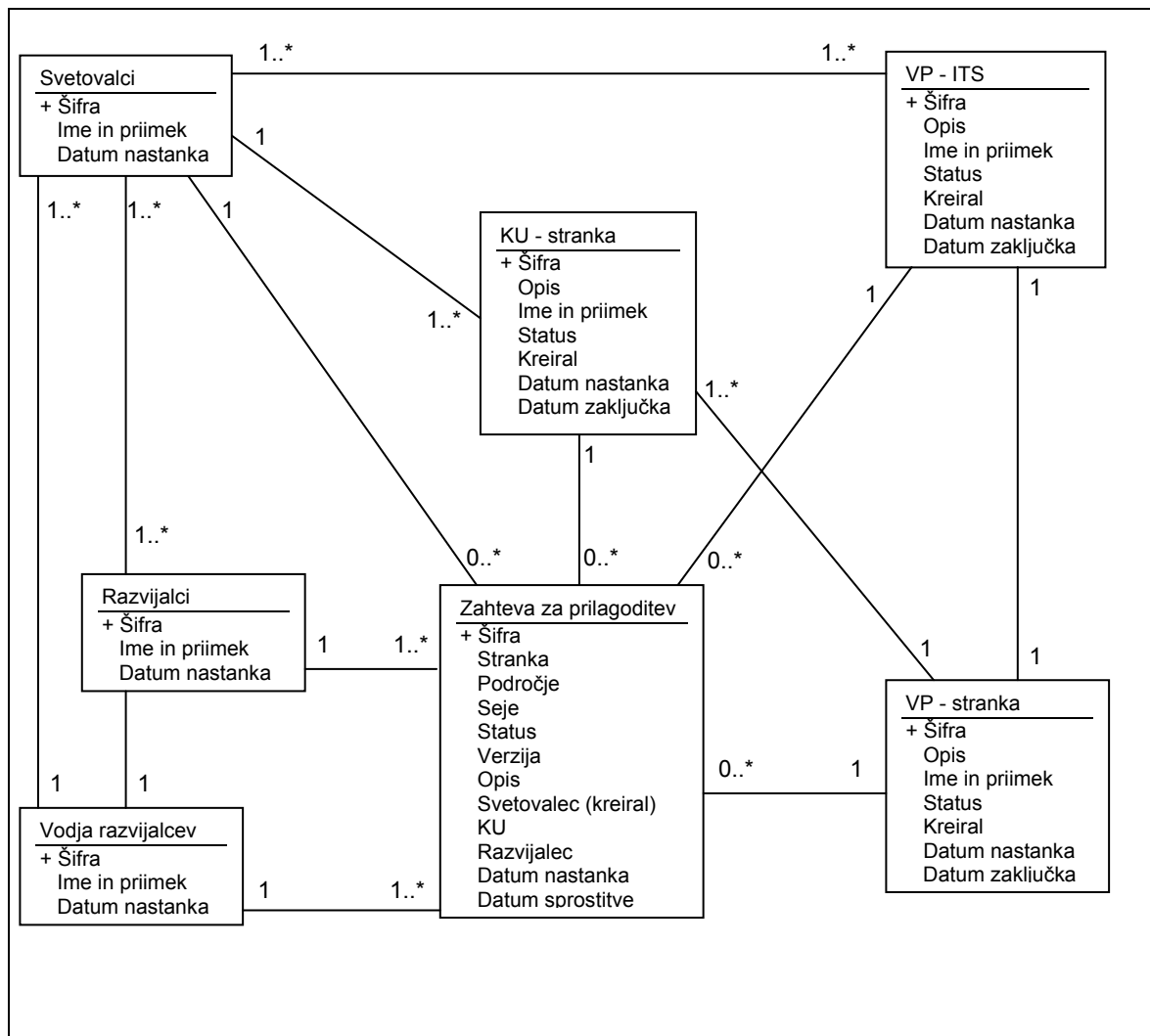


Diagram 10 prikazuje razredni diagram in povezave med razredi. Ta diagram bo osnova za določevanje obrazcev in polj na obrazcih, ki jih lahko v aplikaciji enostavno prilagajamo. Iz diagrama je razvidno kako so razredi med seboj povezani (npr. vsaka zahteva za prilagoditev je povezana z natanko enim svetovalcem, vsak svetovalec pa lahko ima nobeno ali več zahtev za prilagoditev).

Diagram 10: Razredni diagram



5.4 Aktivnostni diagrami

Aktivnostni diagrami opisujejo funkcije sistema. Namenjeni so modeliranju dinamične komponente sistema, uporabljajo pa se za prikaz akcij, ki se izvajajo med določeno operacijo, za prikaz akcij, ki se izvajajo znotraj objekta ali pa za prikaz, kako se izvaja določen primer uporabe.

Aktivnostni diagrami so precej podobni diagramom pretoka podatkov. Z njim običajno podrobneje opišemo korake realizacije kakšne kompleksnejše operacije. Namen diagrama je osredotočenje na tokove notranjih procesov. Medtem ko diagrame stanj uporabljamo v primeru asinhronskih dogodkov, pa aktivnostne diagrame uporabljamo ko večina dogodkov predstavlja zaključek akcij, nastalih znotraj objektov.

Diagram 11: Aktivnostni diagram

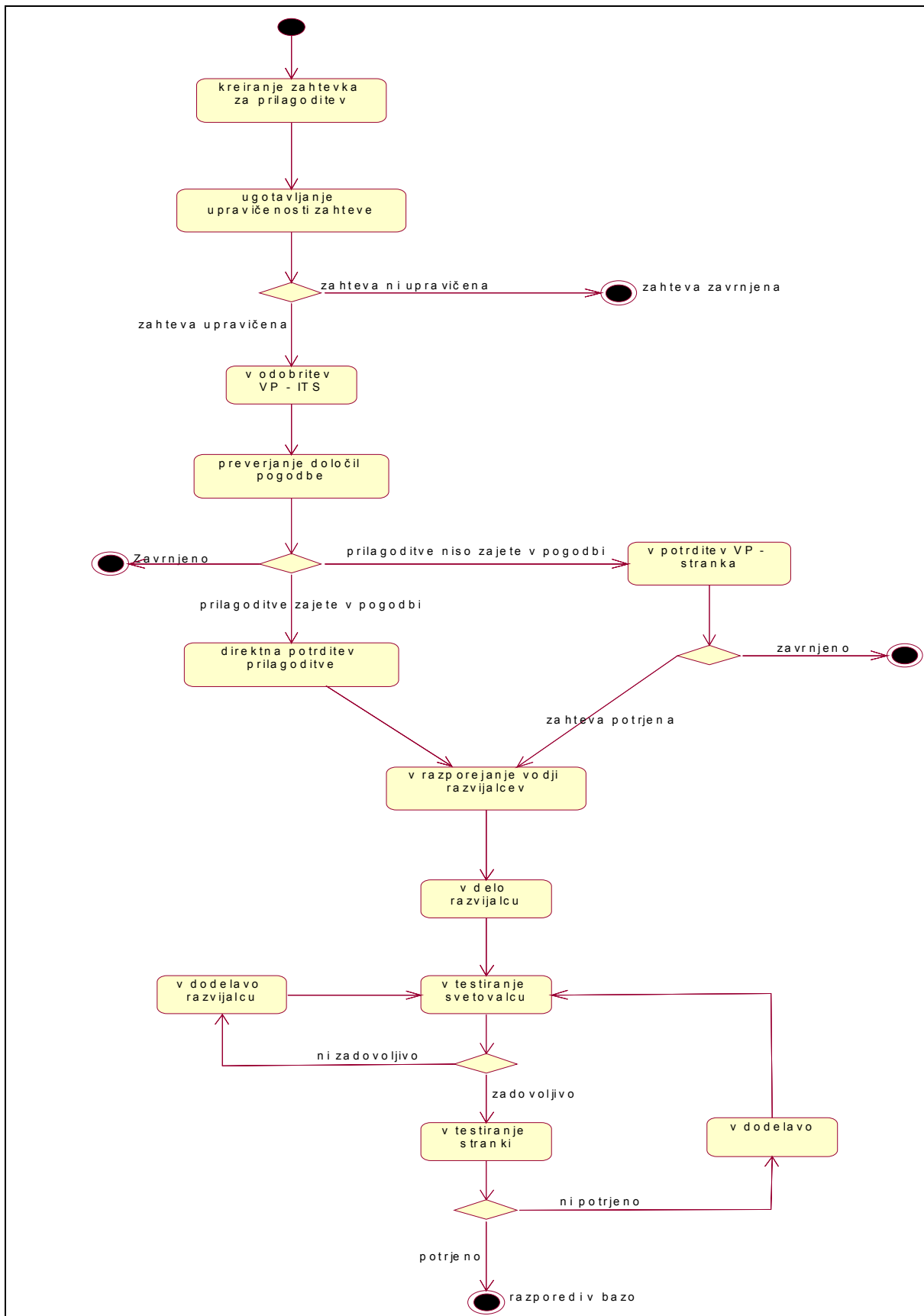


Diagram 11 prikazuje kako se izvajajo primeri uporabe in korake, ki jih moramo izvesti, da izpeljemo določeno prilagoditev do konca. V primeru uvajanja obstoječega PDM programskega paketa je uporaba aktivnostnega diagrama za opis primerov uporabe najprimernejša, saj nam najboljše in najnazorneje opisuje zahtevano delovanje sistema.

Takšen aktivnostni diagram je lahko dobra osnova za svetovanje stranki pri uvajanja sistema, saj omogoča stranki lažjo predstavo o natančnem delovanju sistema in s tem hitrejšo in kvalitetnejšo uvajanje. Ta diagram je tudi v veliko pomoč svetovalcu pri prepoznavanju potrebnih statusov posameznih vrst dokumentov ali map. Prav tako se zmanjša možnost napak pri nastavitvi sistema, saj omogoča sistematično delo.

6. Uvajanje in nastavitve PDM sistema s pomočjo UML

V prejšnjem poglavju sem podrobno analiziral pretok in usmerjanje dokumentacije v razvojnem oddelku podjetja ITS. Na trgu je mnogo PDM in ERP produktov. V podjetju ITS se ukvarjamo z uvajanjem PDM sistema Profidb in ERP sistema Baan v podjetja. Zaradi tega in znanja o produktih, ki je na razpolago, je smiselno uporabiti tudi ta dva sistema za interno uporabo.

Osnova za pričetek nastavitve sistema je podrobna analiza želenega in obstoječega stanja. V našem primeru bom uporabil rezultate analize z UML. Na podlagi diagramov in tabel bom identificiral vrste dokumentov, ki jih bomo uporabili, mape in statute posameznih map in vrst dokumentov.

6.1 Definicija delovnih map v PDM sistemu

Če želimo nastaviti avtomatsko usmerjanje dokumentacije in avtomatsko obveščanj, mora vsak uporabnik v PDM sistemu imeti svojo delovno mapo. Delovne mape najlažje definiramo s pomočjo diagrama primerov uporabe (diagram 1). Iz tega je razvidno kdo so uporabniki oziroma vrste uporabnikov, ki uporabljajo sistem. Vsak uporabnik bo potreboval svojo delovno mapo.

Vrste delovnih map ugotovljene s pomočjo akterjev iz diagrama primerov uporabe in razrednega diagrama:

1. Delovne mape »Stranka – ključni uporabnik (KU)«: Stranke večinoma nimajo dostopa do internega sistema za upravljanje z dokumentacijo. Vsa komunikacija s ključnimi uporabniki stranke poteka preko svetovalcev, ki so zadolženi za določeno področje pri določenem projektu. Zaradi tega je tudi

smiselno, da bo s to mapo upravljal svetovalec, ki komunicira z uporabnikom. Zaradi preglednosti in poenostavitve iskanja mape lahko to mapo povežemo kot podmapo svetovalčeve delovne mape. Za kreiranje teh map in njihove povezave je zadolžen vsak svetovalec za svoje stranke. To pomeni, da bo vsak svetovalec imel v svojo delovno mapo povezanih toliko delovnih map KU – stranka, kot ima projektov v delu.

2. Delovne mape »Svetovalec«: Svetovalci so interni uporabniki sistema. Teh delovnih map mora biti toliko, kot je svetovalcev v podjetju.
3. Delovne mape »Vodja projekta (VP) – ITS«: Vsak svetovalec v podjetju je lahko vodja projekta pri določeni stranki in vsak vodja projekta je lahko samo svetovalec pri drugem projektu. Za potrebe projektnega vodenja kreiramo za vsakega svetovalca, ki je tudi projektni vodja, delovno mapo tipa projektni vodja. Vsak svetovalec bo torej imel v svoji delovni mapi podmape tipa projektni vodja za vsak projekt, katerega vodja je.
4. Delovne mape »Vodja projekta (VP) – stranka«: Podobno kot za ključne uporabnike velja, da vodje projekta pri strankah nimajo direktnega dostopa do internega dokumentacijskega sistema. Komunicirajo z VP – ITS in zato je smiselno, da tudi njihovo delovno mapo upravlja VP – ITS. Vsak VP – ITS bo torej kreiral za vsak projekt dodatno delovno mapo tipa VP – Stranka in jo povezal v svojo delovno mapo.
5. Delovne mape »Vodja razvijalcev«: V podjetju obstaja samo en vodja razvijalcev, ki usmerja delo v razvoju. Zato bo v sistemu ena delovna mapa tega tipa.
6. Delovne mape »Razvijalci«: Razvijalcev je v podjetju več in vsak od njih bo moral imeti svojo delovno mapo v katero se bodo usmerjali dokumenti.

V diagramu primerov uporabe najdemo še dva dodatna akterja (baza gotovih in baza zavrnenih prilagoditev). To niso osebe, ki bi komunicirale s sistemom in zato ne potrebujemo dodatnih delovnih map tega tipa. Predstavljajo dokumente tipa zahteve za prilagoditve, ki imajo status zavrnjeno ali sprejeto. Po teh statusih in s pomočjo iskanja med temi tipi dokumentov lahko iščemo po obeh bazah.

Vsaki vrsti delovnih map lahko nastavimo svoj šifrant in način šifriranja ter vklopimo ali izklopimo vodenje statusov. V kolikor se odločimo za vodenje statusov, moramo nastaviti tudi vrste statusov in pravice spreminjanja statusov. Vsaki vrsti delovnih map lahko nastavimo posebne obvezne in neobvezne podatke na obrazcu.

Na podlagi potreb določimo podatke na obrazcih, statuse in način šifriranja (polja lahko definiramo s pomočjo razrednega diagrama):

1. Delovne mape »Stranka – ključni uporabnik (KU)«

Glede na to da so te vrste map vodene preko svetovalčevih map, je najprimernejši način šifriranja s pomočjo govoreče šifre, sestavljene iz imena stranke in področja, za katerega je zadolžen KU (npr. Podjetje X-proizvodnja).

Predvidena polja:

- Šifra
- Opis
- Ime in priimek
- Status (izbirno polje z možnostmi izbire med »aktiven« in »zaključen«)
- Kreiral
- Datum nastanka
- Datum zaključka

Glede na to, da so te vrste map prehodnega značaja in se aktivirajo ob pričetku projekta, ter zapirajo ob zaključku projekta, je smiselno vključiti vodenje statusov. Predvidimo lahko nastavek avtomatske povezave mape v delovno mapo uporabnika, ki je mapo kreiral. Ob spremembi statusa v »zaključen« se ta povezava avtomatsko odstrani.

2. Delovne mape »Svetovalec«

Zaradi funkcionalnosti PDM sistema, je potrebno te vrste delovnih map šifrirati z enako šifro, kot je uporabniško ime osebe, ki jo uporablja. Ko se uporabnik prijavi v sistem s tem uporabniškim imenom, ga avtomatsko postavi v njegovo delovno mapo. Vodenja statusov za to vrsto delovnih map ne planiramo, saj menjavanje teh map ni prehodnega značaja.

Predvidena polja:

- Šifra (uporabniško ime)
- Ime in priimek
- Datum nastanka

3. Delovne mape »Vodja projekta (VP) – ITS«:

Te vrste map so podobno kot mape KU – stranka prehodnega značaja. Ob zaključku vsakega projekta se zaprejo in odstranijo iz mape svetovalca, ki je vodja določenega projekta. Način šifriranja je tudi v tem primeru lahko govoreča šifra, ki vsebuje ime stranke (npr. VP – podjetje X)

Predvidena polja:

- Šifra
- Opis
- Ime in priimek
- Status (izbirno polje z možnostmi izbire med »aktiven« in »zaključen«)
- Kreiral
- Datum nastanka
- Datum zaključka

Glede na to, da so te vrste map prehodnega značaja in se aktivirajo ob pričetku projekta, ter zapirajo ob zaključku projekta, je smiselno vključiti vodenje statusov. Predvidimo lahko nastavitev avtomatske povezave mape v delovno mapo uporabnika, ki je mapo kreiral. Ob spremembi statusa v »zaključen« se ta povezava avtomatsko odstrani.

4. Delovne mape »Vodja projekta (VP) – stranka«

Te vrste delovnih map so povezane v delovne mape »VP – ITS«. Imajo enake značilnosti in zato lahko predvidimo tudi enaka polja in statute.

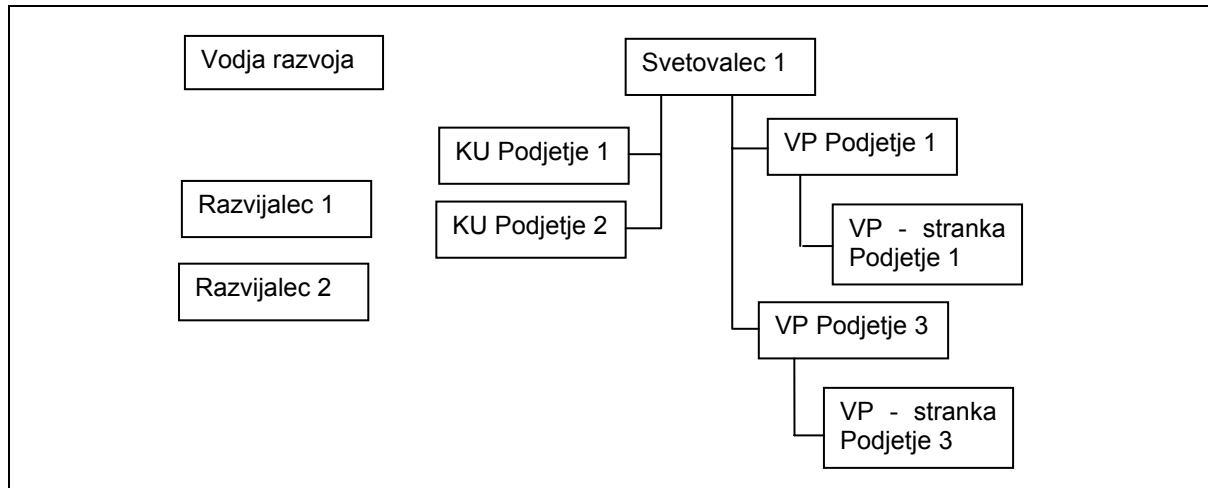
5. Delovne mape »Vodja razvijalcev«

Ta vrsta delovnih map ima enake značilnosti kot delovne mape svetovalcev, zato lahko predvidimo enaka polja, način šifriranja in vodenje statusov.

6. Delovne mape »Razvijalci«

Za delovne mape razvijalcev velja enako kot za delovne mape svetovalcev in vodje razvijalcev.

SLIKA 29: Struktura delovnih map v PDM sistemu



6.2 Definicija projektnih map in njene strukture

Zaradi preglednosti in zbiranja podatkov, je nujno potrebno da vso dokumentacijo, ki je povezana s projekti, zberemo skupaj na enem mestu. To lahko dosežemo s pomočjo projektnih map in podmap. Rezultat je drevesna struktura projekta. Odpre in upravlja jo vodja projekta v ITSu.

Osnovne podmape kreiramo po področjih (proizvodnja, finance...). Kreiramo tudi dodatne podmape za potrebe vodenja projekta (nadzorni odbori, računi, ponudbe...). Primer strukture projektne mape je prikazan na sliki 14. Vsaka podmapa lahko vsebuje poljubno število dodatnih podmap v različnih nivojih.

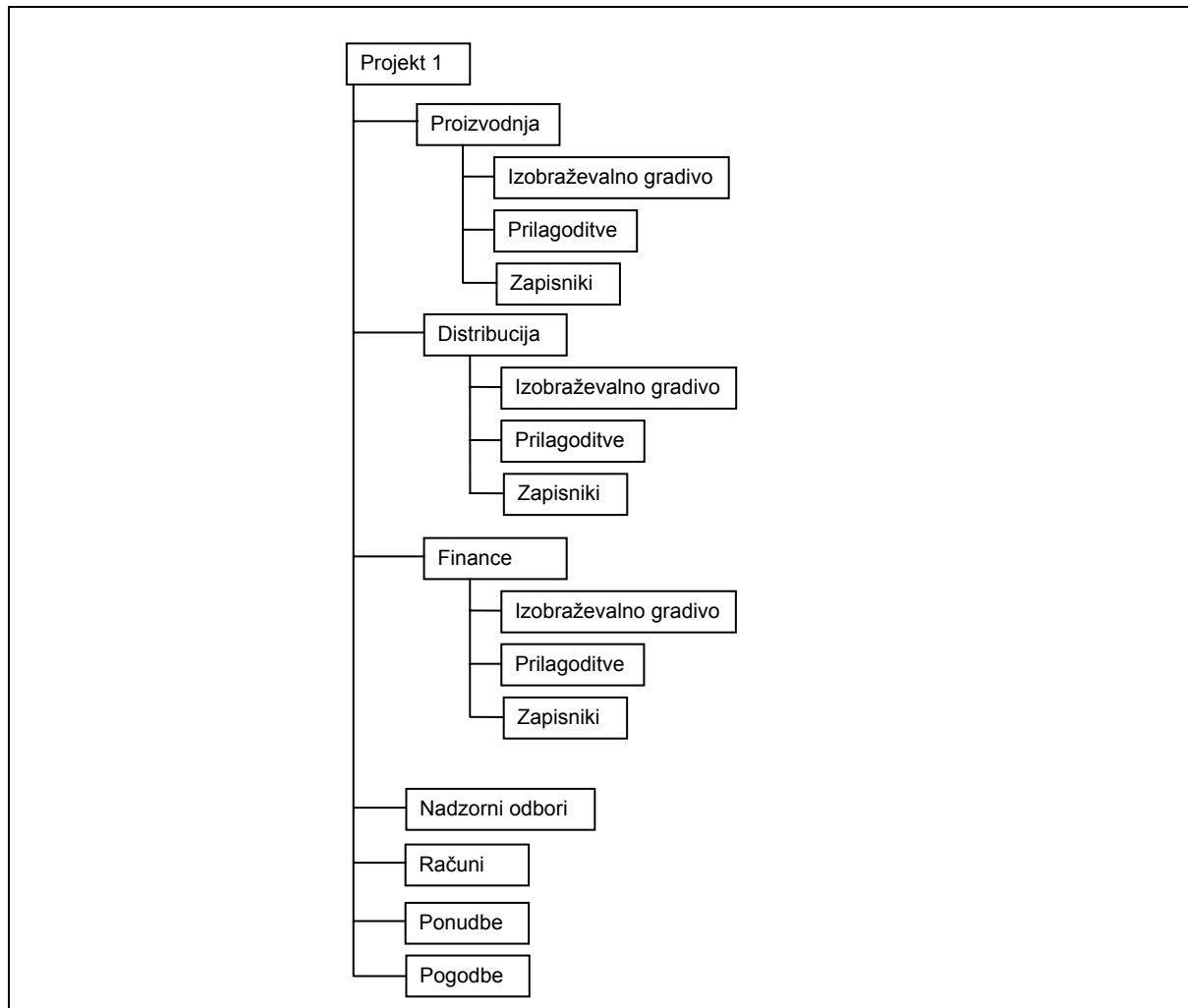
Z enostavno dodatno prilagoditvijo programske opreme je možno v sistemu nastaviti avtomatsko odpiranje osnovne strukture podmap. Glede na specifičnost posameznih projektov, lahko svetovalci ali vodje projektov osnovno strukturo po potrebi dograjujejo. Seveda je potrebno nastaviti tudi pravice spreminjanja strukture in statusov projektni map in njenim podmapam.

Glede na potrebe projektnega vodenja lahko definiramo statuse projektni map, ki ustrezajo osnovnim fazam projekta po Baanovi target metodologiji:

- V delu
- Projektni plan izdelan
- Izobraževanje 1
- Simulacija 1
- Prilagoditve definirane
- Izobraževanje 2

- Simulacija 2
- Prilagoditve izdelane
- Podatki preneseni
- Simulacija 3
- V živo
- Optimizacija

Slika 30: Struktura projektne mape



Osnovno projektno mapo ima pravico spreminjati samo vodja projekta, medtem ko posamezne podmape lahko spreminjajo svetovalci področij. Temu primerno nastavimo pravice v sistemu. Svetovalci posameznih področij lahko spreminjajo dokumentacijo, ki je v njihovih podmapah, vso ostalo pa lahko samo pregledujejo. Da lahko dosežemo ustrezne nastavitve pravic, rabimo določene podatke na obrazcih map in podmap.

Predvidena polja na projektni mapi:

- Podjetje (šifra)
- Opis
- Naslov
- Telefon
- Fax
- Vodja projekta
- Datum začetka
- Datum zaključka
- Status

Predvidena polja na podmapah so enaka. Področne podmape imajo namesto vodje projekta navedenega svetovalca za področje. Ostale podmape so bolj preglednega značaja in ne potrebujejo statusnega vodenja. Na obrazcu so samo osnovna polja (šifra, opis, datum kreiranja in kdo jo je kreiral).

6.3 Definicija dokumentov in vrst dokumentov

V primeru razvoja prilagoditev se uporabljajo predvsem MS Wordovi dokumenti. Z UML pristopom smo opisali tok dokumenta »Zahteva za prilagoditev«. V procesu sodelujejo tudi ostali dokumenti kot so specifikacija, prevzemni zapisnik...

Dokumente, ki se lahko kreirajo na osnovi predloga in imajo svoj krog usmerjanja je smiselno uvrstiti med posebne vrste dokumentov. PDM sistem nam omogoča definicijo poljubno mnogo vrst dokumentov, ki nastajajo z določeno aplikacijo (imamo lahko več vrst wordovih, excelovih in ostalih dokumentov). Vsaka vrsta dokumentov ima lahko svoj obrazec, statuse in način šifriranja. V našem primeru bomo uporabili dve vrsti wordovih dokumentov. Zahteve za prilagoditev usmerjamo na drugačen način kot ostale dokumente, ki jih kreirajo vsi uporabniki posamezno in nimajo določenega delovnega poteka.

Sistem omogoča vezavo dokumentov, ki imajo povezano datoteko ali pa ne. V sistem lahko zavedemo dokument brez datoteke in npr. opišemo mesto hranjenja.

Obrazec vrste dokumentov »Zahteva za prilagoditev« mora vsebovati vsa polja, ki jih potrebujemo za kasnejše iskanje zahtevkov in za avtomatski prenos podatkov v predlogo.

Polja na obrazcu vrste dokumentov »Zahteva za prilagoditev«:

- Šifra (govoreča in vsebuje področje in zaporedno številko)
- Stranka
- Področje
- Seje
- Status
- Verzija
- Opis
- Svetovalec (kreiral)
- KU
- Razvijalec
- Datum nastanka
- Datum sprostitve

Statusi vrste dokumentov »Zahteva za prilagoditev« popisujejo fazo v kateri se zahtevek nahaja. PDM sistem omogoča usmerjanje dokumentov in elektronsko obveščanje med posameznimi fazami.

Na podlagi primera razdelanega z UML pristopom lahko najlaže s pomočjo diagramov primerov uporabe in aktivnostnih diagramov definiramo status dokumentov.

Statusi vrste dokumentov »Zahteva za prilagoditev«:

- V delu
- Potrjeno svetovalec
- Zavrnjeno svetovalec
- Zavrnjeno VP – ITS
- V potrditev VP – stranka
- Zavrnjeno stranka
- Potrjeno VP
- Razporejeno
- Testiranje svetovalec
- Dodelava
- Testiranje stranka
- Potrjeno

Vse ostale dokumente vrste wordovi dokumenti uvrstimo med splošne wordove dokumente, kjer potrebujemo na obrazcu samo osnovne podatke o datumih nastanka, uporabnikih, ki so jih kreirali, verzijami in opisi dokumentov.

6.4 Elektronsko usmerjanje dokumentacije in obveščanje

V PDM sistemu usmerjamo dokumentacijo s pomočjo definicije akcij, ki se izvedejo ob spremembah statusov.

V prejšnjem poglavju smo s pomočjo UML diagramov primerov uporabe in aktivnostnih diagramov definirali statute dokumentov vrste »Zahteva za prilagoditev«. Isti diagrami bodo pomagali definirati tudi akcije, ki se izvedejo ob spremembah teh statusov. Akcija se mora izvesti vsakič, ko želimo nekoga obvestiti o spremembi statusa oziroma usmeriti dokument v novo mapo.

Elektronsko usmerjanje in obveščanje ob spremembah statusov zahteve za prilagoditev:

1. Iz statusa »v delu« v status »potrjeno svetovalec«: Ob tej akciji je iz omenjenih UML diagramov razvidno, da je naslednji akter v procesu VP – ITS. Sistem moramo torej nastaviti tako, da se ob tej spremembi statusa zahteva za prilagoditev avtomatsko poveže v delovno mapo VP – ITS. Istočasno se mora kreirati tudi elektronsko sporočilo VP – ITS o novi zahtevi, ki jo mora obdelati. Glede na to, da je lastnik prilagoditve svetovalec, zahteva še vedno ostane povezana tudi v njegovi mapi.
2. Iz statusa »potrjeno svetovalec« v status »zavrnjeno«: VP – ITS lahko zahtevo po prilagoditvi direktno zavrne in spremeni temu primerno status. Ob tej spremembi statusa se mora izbrisati povezava dokumenta v delovni mapi VP – ITS. Svetovalcu pa se mora poslati avtomatsko kreirano elektronsko sporočilo, ki ga obvesti o zavrnitvi. Sistem omogoča dodaten vnos razloga za zavrnitev pred pošiljanjem obvestila.
3. Iz statusa »potrjeno svetovalec« v status »potrjeno«: V tem primeru je VP – ITS ugotovil, da je prilagoditev zajeta v pogodbi in ni potrebna odobritev stranke. Dokument se avtomatsko izbriše iz delovne mape VP – ITS in direktno poveže v delovno mapo vodje razvijalcev. Temu primerno se kreira tudi avtomatsko elektronsko sporočilo svetovalcu in vodji razvijalcev.
4. Iz statusa »potrjeno svetovalec« v status »V potrditev VP – stranka«: Prilagoditev ni zajeta v pogodbi in mora v potrditev stranki. Kreira se avtomatsko elektronsko sporočilo s pripeto datoteko, ki ga VP – ITS opremi s komentarji in pošlje VP – stranka. Svetovalec prav tako dobi avtomatsko

kreirano elektronsko obvestilo o dogajanju. Zaradi boljše preglednosti se zahteva prestavi v podmapo VP – stranka.

5. Iz statusa »V potrditev VP – stranka« v status »potrjeno«: Dokument se avtomatsko izbriše iz delovne mape VP – stranka in direktno poveže v delovno mapo vodje razvijalcev. Temu primerno se kreira tudi avtomatsko elektronsko sporočilo svetovalcu in vodji razvijalcev.
6. Iz statusa »potrjeno« v status »Razporejeno«: Vodja svetovalcev razporedi izdelavo prilagoditve v delo razvijalca. Na obrazcu mora izbrati razvijalca in spremeniti status, kar omogoči avtomatsko povezavo dokumenta v delovno mapo izbranega razvijalca, ki prejme tudi avtomatsko kreirano obvestilo. Kopija obvestila se pošlje tudi svetovalcu, ki mora biti ves čas obveščen o stanju prilagoditve.
7. Iz statusa »Razporejeno« v status »testiranje svetovalec«: Razvijalec po končnem delu spremeni status in izbriše se povezava zahteve iz razvijalčeve mape. Svetovalec dobi avtomatsko kreirano elektronsko obvestilo.
8. Iz statusa »testiranje svetovalec« v status »dodelava«: Svetovalec ob testiranju ugotovi potrebne dodelave. Zahteva se ponovno poveže v delovno mapo razvijalca, ki prejme tudi avtomatsko obvestilo, ki ga svetovalec dopolni z navodili pred pošiljanjem.
9. Iz statusa »testiranje svetovalec« v status »testiranje stranka«: Ko je prilagoditev potrjena s strani svetovalca, jo le ta pošlje v testiranje k stranki. Kreira se avtomatsko obvestilo stranki, zaradi boljše preglednosti pa se tudi zahteva avtomatsko prestavi v podmapo KU pri svetovalcu.
10. Iz statusa »testiranje stranka« v status »dodelava«: Stranka ugotovi nepravilnosti in prilagoditev se mora vrniti v dodelavo k razvijalcu. Avtomatsko se kreira obvestilo razvijalcu, ki ga svetovalec opremi z navodili. Zahteva se avtomatsko prestavi nazaj v delovno mapo svetovalca in dodatno poveže v delovno mapo razvijalca.
11. Iz statusa »testiranje stranka« v status »potrjeno«: Stranka je prilagoditev sprejela. Kreira se avtomatsko obvestilo VP – ITS, zahteva pa se prestavi nazaj v delovno mapo svetovalca (razporejeno v projektno podmapo prilagoditev).

V PDM sistemu Profidb administratorsko orodje omogoča enostavno nastavljanje akcij ob spremembah statusov. S pomočjo UML diagramov pa smo lahko hitro in pregledno definirali potrebne spremembe.

6.5 Vodenje sprememb in sledljivost

Zaradi vrednotenja stroškov in uspešnosti projekta moramo nujno spremljati število sprememb in dodelav prilagoditev pred doseženim končnim stanjem. V PDM sistemu imamo več možnosti za zagotavljanje sledljivosti.

Sistem moramo nastaviti tako, da se med dokumentne opombe (dodatna povezava k vsakemu dokumentu) beležijo vse spremembe statusov. Nastavimo lahko tudi obvezen vnos opomb ob spremembah v določene statuse, kamor se mora vpisati razlog za spremembo statusa. V našem primeru je to smiselno definirati ob vsaki spremembi v status »dodelava«. Na ta način lahko kadarkoli vidimo koliko dodelav je bilo izvedenih na posamezno prilagoditev in kakšni so bili razlogi za dodelavo.

Po odobritvi prilagoditve s strani VP – ITS zahteve ne smemo več spreminjati. To kontrolo lahko definiramo v PDM sistemu s pomočjo pravic o spreminjanju dokumentov ob določenih statusih, ki se nastavijo v administratorskem orodju aplikacije. V kolikor želimo spremeniti zahtevo, ko je le ta že v potrjenem ali kasnejšem statusu, lahko sistem nastavimo tako, da je to mogoče le ob kreiranju nove verzije. Obvezno moramo navesti vzrok kreiranja nove verzije, ki mora ponovno čez postopek odobritve.

Na ta način dobimo s pregledovanjem verzij dokumentov (zahtev za prilagoditev) in opomb, ki so nastale ob spreminjanju statusov ali kreiranju novih verzij, popolno sledljivost o tem, kaj se je dogajalo s prilagoditvijo od nastanka zahtevka pa vse do potrditve s strani stranke. Vidimo lahko število in vzroke dodelav, kar lahko razjasni mnoge vzroke morebitnih zapletov.

7. Povezovanje med PDM in ERP sistemom Baan v podjetju

Da bi lahko predvideli povezovanje PDM sistema na ERP sistem, moramo najprej predvideti način dela v ERP sistemu in temu primerno nastaviti PDM sistem in integracijo.

V ITS-u uvajamo ERP sistem Baan. Delo je projektne narave, zato je za vodenje projekta uvajanja smiselno uporabiti Baanov modul projekti. Uvajanje projektne modula za delo v podjetju ITS trenutno ni v kratkoročnem planu. Prav tako obseg uporabe še ni definiran. V nadaljevanju sem prikazal eno izmed možnih rešitev, ki

pa jo bo potrebno s časom dopolniti. Najprej bo v uporabi PDM sistem s predvidenimi možnostmi povezav v ERP sistem. Med testiranjem pilotskega modela PDM sistema in delom »v živo« bomo prišli do novih ugotovitev in potreb po spremembah nastavitvev. Po končanem procesu uvajanja PDM sistema bo šele mogoče pripraviti natančen načrt uvajanja celotnega projektnega modula ERP sistema Baan in natančen načrt povezovanja obeh sistemov.

Trenutno je v uporabi okrnjena oblika dela v ERP sistemu Baan, ki je namenjena spremljanju porabljenega dela na posameznem projektu in služi kot osnova za zaračunavanje. Uporabljena rešitev bo kasneje verjetno integrirana v uporabi celotnega projektnega modula in kot samostojna ni primerna za povezovanje s PDM sistemom.

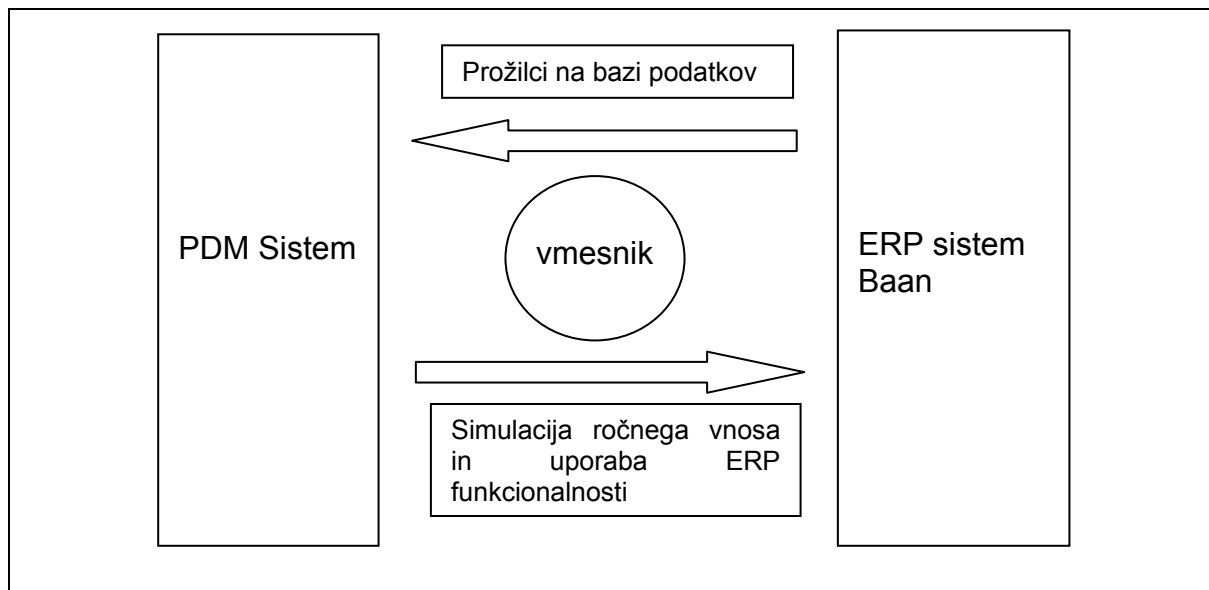
7.1 Projektno vodenje v Baan ERP sistemu

Baanov modul projekti omogoča celotno spremljanje projektov od predračunskih vrednosti, ponudb do pokalkulacij zaključenih projektov. Za vodenje projektne dokumentacije je smiselno kot nadgradnjo uporabiti PDM sistem.

Za komunikacijo med PDM sistemom in Baan ERP sistemom (smer iz PDM sistema v ERP sistem) skrbi vmesnik, ki popravlja oziroma briše in dodaja podatke v ERP sistem s pomočjo simuliranja ročnega vnosa podatkov. Na ta način dosežemo preverjanje vsakega vnesenega podatka s kontrolami, ki so vgrajene v ERP sistem. V kolikor bi hoteli podatke vnašati direktno v bazo podatkov ERP sistema, bi morali vse te kontrole ponovno vgraditi v vmesnik, kar pa ni smiselno. Na ta način se lahko izognemo tudi vsem nepravilnostim, ki bi nastale v primeru takšnega povezovanja ob posodobitvah ERP sistema. Če so kontrole vgrajene v vmesnik med sistemoma, bi morali ob vsaki posodobitvi ERP sistema posodobiti tudi vmesnik, saj bi drugače prihajalo do napak. V primeru simulacije ročnega vnosa pa vsaka sprememba kontrol podatkov v ERP sistemu nemudoma avtomatsko vpliva tudi na delovanje vmesnika.

Na sliki 31 vidimo shematski prikaz delovanja vmesnika. V smeri prenosa podatkov iz ERP sistema v PDM sistem je najenostavnejša in najzanesljivejša pot z uporabo prožilcev na bazi podatkov. Pogoji za to je, da oba sistema uporabljata isto bazo podatkov (npr. SQL ali Oracle). Glede na to, da gredo v tej smeri samo podatki in se ne izvajajo procedure je takšna pot povsem zadovoljiva in zanesljiva. V primeru različnih baz podatkov, moramo uporabiti dodatne funkcionalnosti vmesnika, ki pa delujejo počasneje in zahtevajo večji nadzor.

SLIKA 31: Prikaz delovanja vmesnika med ERP sistemom Baan in PDM sistemom.



Projektne aktivnosti se začnejo s poslovnimi priložnostmi in kreiranjem projektov tipa predračun (budget). S pomočjo ocenjenih stroškov tega projekta v Baan ERP sistemu direktno kreiramo ponudbo, ki jo lahko kasneje popravljamo, stranka pa jo sprejme ali zavrne.

V tej točki se kaže prva potreba po integraciji s PDM sistemom, ki skrbi za dokumentacijski del projekta.

Ob kreiranju Projekta v Baanu se preko vmesnika kreira projekt v PDM sistemu. Kreira se projektna mapa in projektni artikel z določenim statusom (npr. v delu). V PDM sistem se prenesejo vsi pomembni podatki iz Baana, kot so poslovni partner, naslov, ocenjeni stroški, procent uspeha...

Glede na to, da v ITSuodobritev ponudbe zahteva podpis več oseb, lahko v PDMu vodja prodajnih projektov kreira ponudbo po prednastavljenih predlogah in jo s spremembo statusa avtomatsko pošlje v pregled vsem podpisnikom.

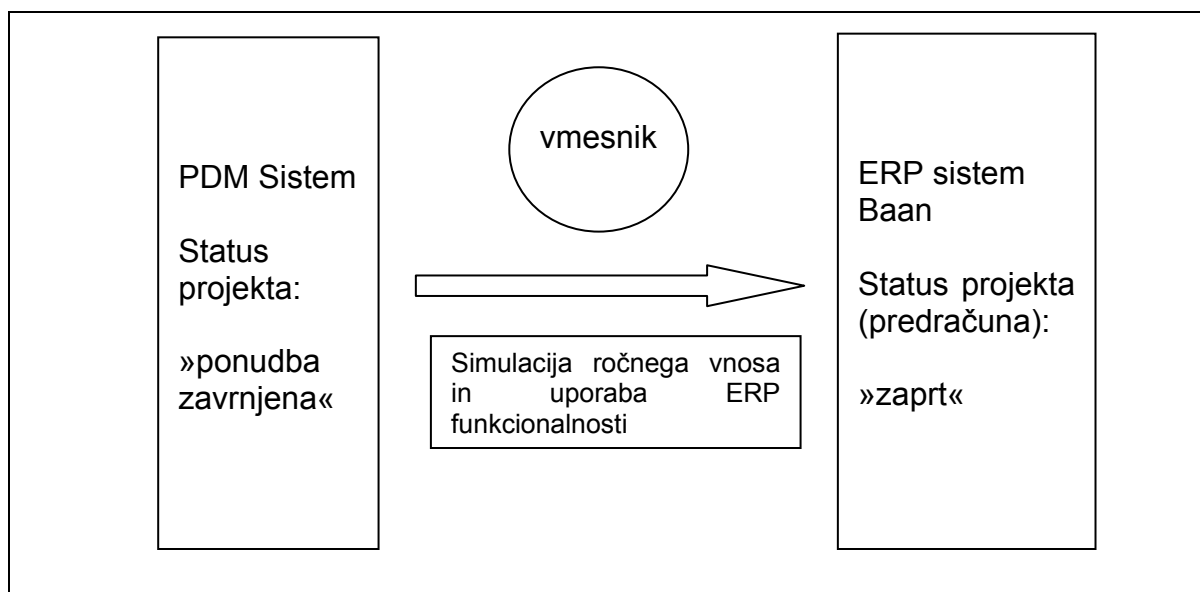
Vsi podpisniki ponudbo potrdijo ali zavrnejo s pomočjo spremembe statusa in PDM sistem temu primerno usmeri dokumente v delovno mapo vodje prodajnih projektov.

Ko je ponudba potrjena s strani odgovornih oseb v ITSu, jo vodja prodajnih projektov pošlje potencialni stranki in na koncu zabeleži rezultat ponudbe.

V projektno mapo se lahko poveže še vsa ostala dokumentacija v zvezi s tem projektom (dokumenti, ki jih je poslala stranka, prezentacije, posnetki stanja...).

V primeru, da ponudba ni bila uspešna vodja prodajnih projektov spremeni status v npr. neuspešno in tu nastopi druga točka integracije z ERP sistemom.

SLIKA 32: Povezava med PDM in ERP sistemom ob neuspeli ponudbi.



V ERP sistemu mora integracijski vmesnik avtomatično zaključiti odprt projekt tipa predračun in v tem scenariju je delo zaključeno.

Ob uspešni ponudbi sledi podpis pogodbe in temu ustrezen status projekta. Tudi na tem mestu moramo vključiti integracijski vmesnik, ki mora v ERP sistemu spremeniti projekt tipa predračun v projekt tipa enonivojski projekt. Ob tej akciji se tudi zamrznejo predračunski stroški, ki kasneje služijo kot informacija ob pokalkulaciji projekta.

V ERP sistemu to ni enostavna akcija spremembe statusa ampak mora vmesnik preko nekaj sej simulirati ročni vnos in zaključiti projekt tipa predračun in ta predračun kopirati v normalen projekt.

Ko v ERP sistemu nastane potrjen projekt sledi sestava projektne dokumentacije, za katero skrbi vodja projekta. Sestaviti mora projektni plan in po projektne planu

vnesti aktivnosti projekta v ERP sistem. MSProject dokument projektnega plana se preko PDM vmesnika z MSProject shrani v PDM sistem v pripadajočo projektno mapo. Temu dokumentu nato spreminjamo status glede na fazo v kateri se nahaja (v delu, oddan, sprejet, zavrnjen...).

Vse aktivnosti in njene časovne ocene se nato prenesejo v projektno mapo preko vmesnika med ERP in PDM sistemom. Vmesnik je potrebno nastaviti tako, da je iz informacij o aktivnostih v PDM sistemu razvidno kdo je odgovoren za posamezno aktivnost in koliko časa je bilo za njo planirano. Vsaka aktivnost je neke vrste mapa v projektni strukturi za katero je odgovoren Baan svetovalec za določeno področje. V te aktivnostne mape je možno pripenjati vso dokumentacijo, ki je v zvezi s to aktivnostjo (npr. posnetki stanja, za stranko prilagojeno izobraževalno gradivo...).

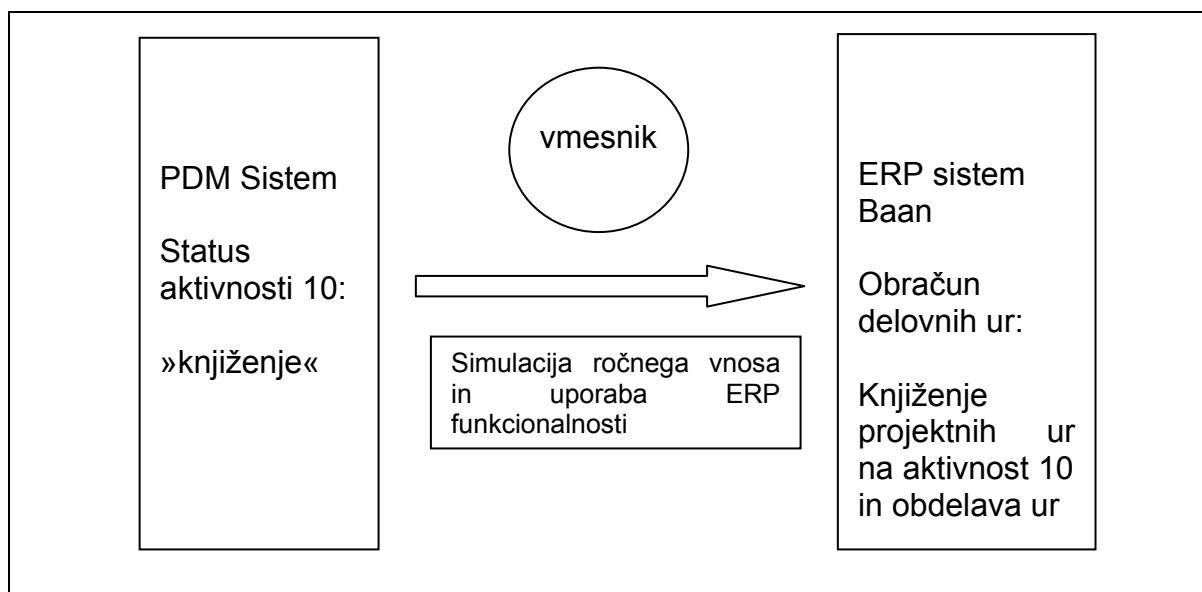
Ko je izdelava projektne dokumentacije zaključena, mora vodja projekta to potrditi s spremembo statusa, kar sproži akcijo spremembe statusa projekta v Baanu. V tem trenutku se v Baanu zamrznejo ocenjeni stroški projekta. Vsaka sprememba projektne dokumentacije od tega trenutka dalje je možna samo s pomočjo nove verzije in vnosa razloga za kreiranje nove verzije. Pravico potrditve nove verzije projektne dokumentacije ima samo vodja projekta. To pomeni da od potrditve projektne dokumentacije dalje spremljamo vse spremembe, ki nastajajo v fazi operativnega uvajanja sistema. Tako imajo vsi, ki sestavljajo projektni tim, vedno dostop do prave verzije projektnega plana.

Po zaključeni aktivnosti svetovalci vnesejo potrebne stroške oziroma porabljeno število dni za izvedbo aktivnosti. Po vnosu vnos potrdimo s spremembo statusa in sledi avtomatsko obveščanje vodje projekta, ki vnese dejansko potrjene časovne stroške s strani stranke. Vodja projekta vnos potrdi s pomočjo spremembe statusa, ki sproži tretjo povezavo med PDM in ERP sistemom.

V tem primeru moramo s pomočjo vmesnika simulirati ročen vnos v Baanove seje za vnos ur po projektnih aktivnostih in v primeru zadnjega vnosa tudi zaključiti aktivnost. Po vnosu ur mora vmesnik sprožiti sejo za obdelavo ur, ki povzroči finančne transakcije.

Dejansko vnesene in obdelane ure na projektnih aktivnostih služijo kot osnova za zaračunavanje.

SLIKA 33: Knjiženje ur na projekt iz PDM sistema v Baan s pomočjo vmesnika.



7.2 Delovanje vmesnika med PDM sistemom in ERP sistemom v razvojni fazi projekta

Med aktivnosti projekta uvajanja ERP sistema spada tudi razvoj prilagoditev in izpisov. Postopek kroženja dokumentacije je razdelan v 6. poglavju. Na tem mestu bi se osredotočili na točke možnosti povezave z ERP sistemom.

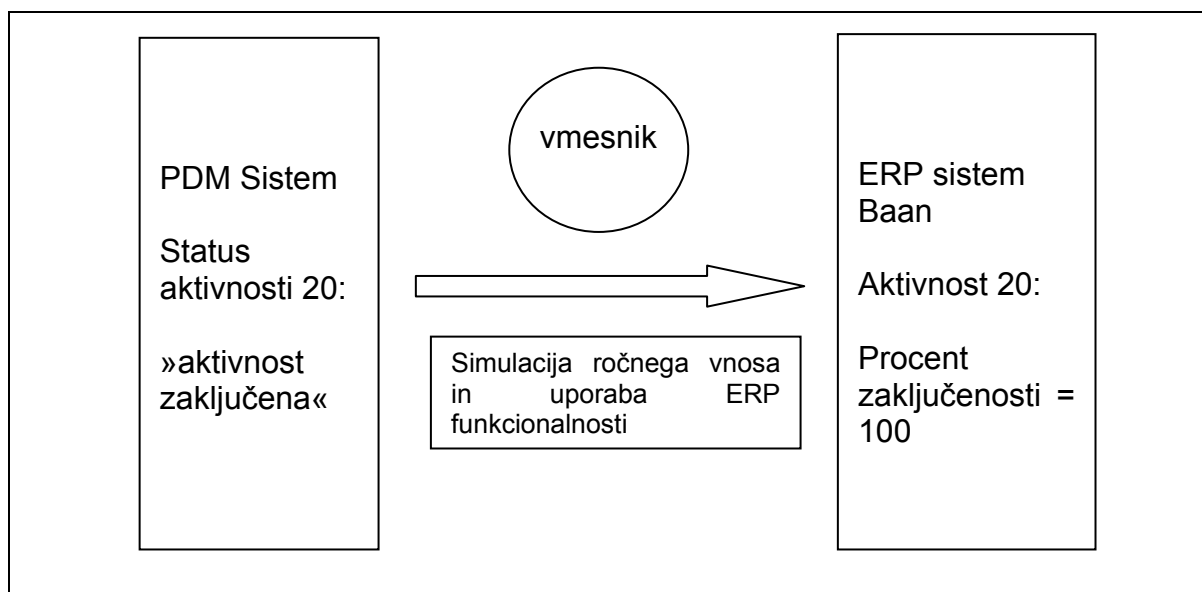
Tako kot vse druge projektne aktivnosti so tudi aktivnosti razvoja prilagoditev časovno in stroškovno vrednotene. Pri aktivnosti razvoja prilagoditev navedemo pri vsaki zahtevi za prilagoditev časovno in posledično stroškovno oceno, ki jo mora stranka potrditi. Ko je zahteva za prilagoditev (ponudba) potrjena s pomočjo spremembe statusa sprožimo vmesnik, ki to oceno vnese med ocenjene stroške aktivnosti. V kolikor ta aktivnost ni bila planirana ob izdelavi projektnega plana mora vmesnik poskrbeti za vnos nove aktivnosti s pomočjo simulacije ročnega vnosa v Banovo sejo.

Ko je stroškovna in časovna ocena aktivnosti izdelave določene prilagoditve potrjena, le te ni več mogoče spreminjati. Ko razvijalec konča z razvojem prilagoditve in svetovalec s testiranjem le te, razvijalec in svetovalec vneseta dejansko porabljeni čas za izdelavo prilagoditve. Svetovalec vnese tudi predlog stroška za zaračunavanje in potrdi aktivnost. S tem sproži avtomatsko obveščanje

vodje projekta, ki vnese po dogovoru s stranko dejanski strošek zaračunavanja in s spremembo statusa potrdi aktivnost.

Ponovno se vključi vmesnik, ki vnese dejanski strošek aktivnosti v ERP sistem in po potrebi aktivnost zaključi (odvisno od statusa aktivnosti v PDM sistemu).

SLIKA 34: Zaključevanje aktivnosti razvoja prilagoditve.



V kolikor se pojavi potreba po dodatnih popravkih prilagoditve po zaključeni aktivnosti, moramo kreirati novo zahtevo za popravek prilagoditve oziroma novo verzijo zahteve za prilagoditev. Po potrditvi izdelave dodatnih zahtev mora vmesnik ponovno poskrbeti za pravilen vnos podatkov v ERP sistem.

7.3 Zaključevanje projekta uvajanja informacijskega sistema

Ko je projekt uvajanja ERP sistema za stranko zaključen, vodja projekta spremeni status projektni mapi v PDM sistemu. Avtomatsko se sproži vmesnik med PDM in ERP sistemom, ki s pomočjo simulacije ročnega vnosa v sejo zapre projekt v ERP sistemu. V tem trenutku se izvede pokalkulacija projekta in izračunajo se dejanski stroški projekta. Glede na to, da je ERP sistem integriran sistem, takrat nastanejo tudi ustrezne finančne transakcije za nadaljnjo obdelavo v računovodstvu.

S pomočjo raznih pregledov in izpisov lahko v ERP sistemu natančno analiziramo predračunske, ocenjene in dejanske stroške projekta in ugotovimo kje so nastala odstopanja. V kolikor je PDM sistem ustrezno nastavljen (npr. obvezne opombe...) lahko ugotovimo tudi vzroke za nastanek odstopanj.

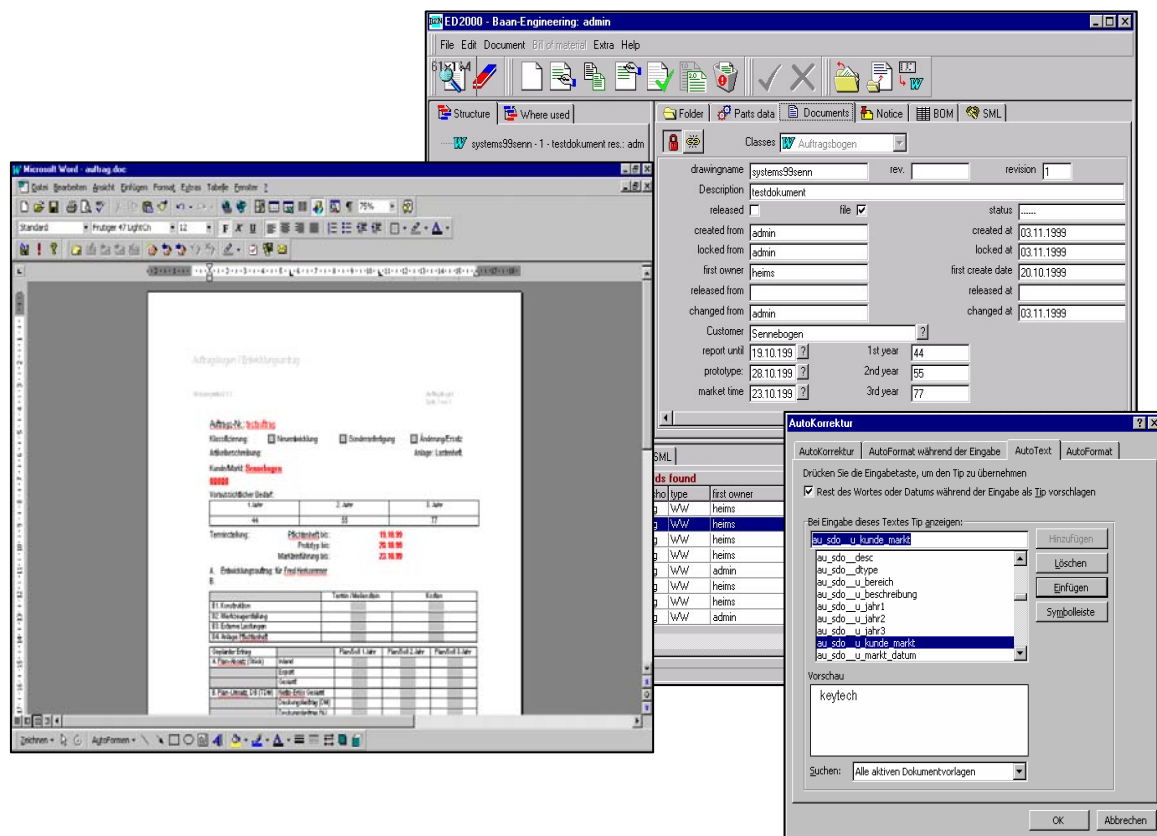
8. Povezovanje med PDM sistemom in MS Office aplikacijami

Velik del projektne dokumentacije nastaja s pomočjo MS Office orodij. PDM sistem oziroma njegov vmesnik omogoča povezovanje z MS office aplikacijami.

Vmesnik v primeru Worda in Excela omogoča:

- nalaganje/hrambo datotek in tekstovnih delov,
- avtomatsko ažuriranje dokumentov,
- uporabo predlog,
- kreiranje avtomatskih tekstov.

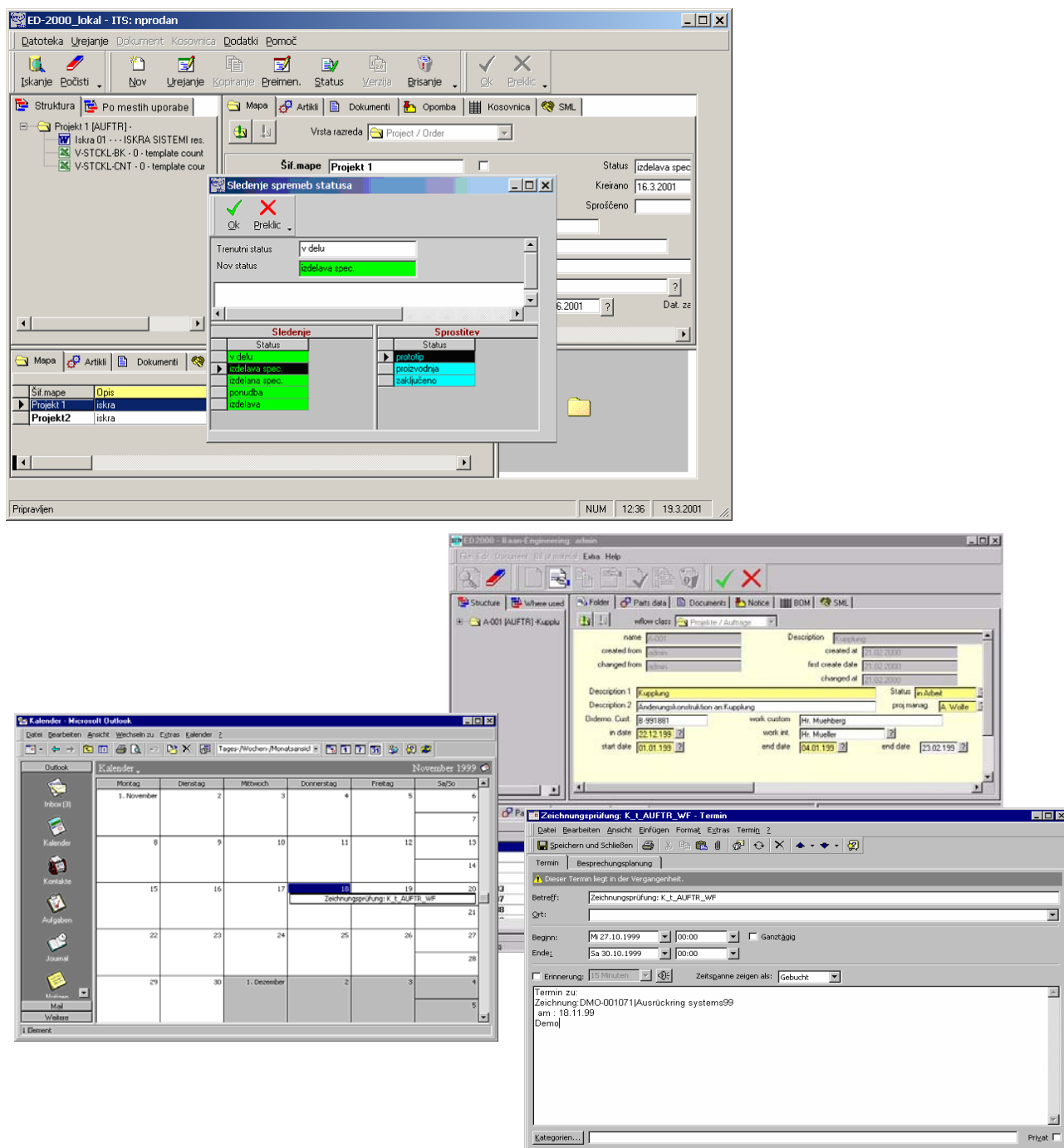
SLIKA 35: Kreiranje dokumentov s pomočjo predlog v Wordu



Na sliki 35 je prikazan primer kreiranja datotek s pomočjo uporabe predlog v MS Wordu. V predlogi preko MS office integracije nastavimo polja, katerih podatki se avtomatsko prenašajo iz forme PDM sistema na določena mesta v Wordovi datoteki. V kolikor predloge dobro nastavimo, nam ta funkcionalnost lahko

omogoči znaten prihranek na času pri kreiranju datotek. Integracija z MS Excel omogoča enake funkcionalnosti.

SLIKA 36: Avtomatsko kreiranje elektronske pošte in MS Outlook sestankov ob spremembi statusa.



V prejšnjih poglavjih smo večkrat omenili možnost avtomatskega obveščanja in elektronskega usmerjanja dokumentacije. Avtomatsko obveščanje lahko razširimo s pomočjo MS Outlook integracije s PDM sistemom. Ta vmesnik omogoča, da se sistem ob določenih spremembah statusov določenih objektov v PDM sistemu avtomatsko poveže z MS Outlook aplikacijo in samostojno pošlje prednastavljena elektronska sporočila določenim osebam. Sporočilo se lahko pošlje popolnoma avtomatsko ali pa sistem ponudi predlog elektronskega sporočila, ki ga lahko dopolnimo z dodatnimi naslovniki in informacijami. Akcije ob spremembah statusov lahko poljubno nastavljamo za različne tipe datotek. Vmesnik omogoča tudi avtomatsko kreiranje MS Outlook sestankov v koledarju ob nastavljenih spremembah statusov objektov v PDM sistemu. V tem primeru so prejemniki in vsebine sestankov prednastavljene in jih ne moremo sproti prilagajati.

9. Sklep

V nalogi sem naredil zasnovo informacijskega sistema za upravljanje z razvojnimi podatki in dokumentacijo izdelkov, ki dopušča veliko prostora za nastavitve. Prišel sem do ugotovitve, da uporaba UML pristopa pri uvajanju takšnega sistema lahko močno olajša delo in omogoči doseganje boljše kvalitete postavljenega sistema. Glede na to, da je takšen sistem že narejena aplikacija, nam pri delu pomagajo predvsem diagrami, ki nazorno in enostavno ponazarjajo poteke dela (diagrami primerov uporabe in aktivnostni diagrami). Za določitev polj na obrazcih nam pomaga definicija razredov in razredni diagram. Vseh diagramov za postavitve takšnega sistema verjetno nima smisla uporabljati, saj ne prinesejo dodatnih koristi, ampak samo povečajo stroške. Seveda pa je primernost uporabe posameznih diagramov odvisna od vrste, kompleksnosti, potreb in zmožnosti programskega paketa in naročnika.

Diagrami primerov uporabe so primerni in uporabni po začetnih intervjujih, saj nam in stranki nazorno prikažejo potek dela v podjetju. Skupen pregled diagramov primerov uporabe s stranko omogoči jasno in nedvoumno sliko o delovnem poteku, kar lahko prihrani veliko kasnejših zapletov, nezadovoljstva in dodatnih stroškov. Izredno pomembno je, da si na začetku uvajanja sistema ustvarimo pravilno sliko poslovanja in da je stranka prepričana, kakšen prenovljen proces si želi. Vizualen potek dogodkov s pomočjo diagramov omogoča, da se na lažji način prepričamo o pravilnosti svetovalčevih ugotovitev. Stranki, ki je kupila takšen sistem, pa nazoren prikaz s pomočjo diagramov omogoči lažje razumevanje in predstavo o delovanju sistema.

Vsako kasnejše popravljanje in dopolnjevanje sistema privede do znatno višjih stroškov uvajanja, nezadovoljstva stranke in ponavadi do manj primernih rešitev, saj se velikokrat kasneje sprejema kompromise zaradi stroškov, ki pa ob pravem razumevanju možnosti sistema s strani stranke in stanja v podjetju s strani svetovalca ne bi nastali.

Torej diagrami primerov uporabe so posebej primerni za izdelavo pravilnega posnetka stanja in omogočajo stranki in svetovalcem neke vrste kontrolo pravilnega razumevanja problemov.

Izredno uporabni so se pokazali tudi aktivnostni diagrami, do katerih pridemo proti koncu dela z UML metodo. V teh diagramih je potek dela še natančneje prikazan in omogoča zadnjo in primerno kontrolo zahtev pred pričetkom nastavitve sistema oziroma programiranjem prilagoditev.

V drugem delu naloge sem prikazal možnost projektnega vodenja sprememb ERP sistema in način ter mesta povezovanja med PDM sistemom in projektnim modulom ERP sistema ter povezovanje med PDM sistemom in MS Office orodji. Poskušal sem poiskati optimalno rešitev povezovanja PDM sistema in ERP sistema za primer natančnega projektnega vodenja. Ob tem sem ugotovil, da vse točke povezave niso zajete v standardnem vmesniku, vendar se bi jih dalo glede na odprtost in možnost dopolnjevanja vmesnika med sistemoma enostavno dograditi.

Iz zgoraj navedenega je razvidno, da je bil cilj magistrske naloge dosežen. Analizo obstoječega stanja dela v razvojnem oddelku, definiranje uporabnikov in postopkov sem prikazal s pomočjo UML pristopa. Ugotovil sem da je ta pristop v okrnjeni obliki primerno uporabljati za prvo fazo načrtovanja informacijskega sistema za podporo razvoju. Prav tako sem v delu prikazal informacijsko infrastrukturo, tehnologije ter funkcionalnosti PDM sistemov. S pomočjo ključnih uporabnikov sem definiral področja uporabe PDM sistema. Glede na postavljene zahteve in tehnologije sem definiral in predlagal module PDM sistema, ki se jih bo uporabilo ob implementaciji. Pripravil sem tudi načrt nadaljnjega razvoja in optimizacije PDM sistema v podjetju ITS intertrade ter možnosti povezovanja z ERP sistemom Baan.

10. Literatura

1. Anderl R., Mendgen R.: *Produktdatentechnologie I, Grundlagen und DV-Systeme, Fachgebiet Datenverarbeitung in der Konstruktion*, Technische Hochschule Darmstadt, 1996, 244 str.
2. Belbin R.M.: *Management Teams, Why They Succeed or Fail*, Butterworth – Heinemann, 1993, 179 str.
3. Bernus P., Nemes L., Williams T.J.: *Architectures for Enterprise Integration*, Chapman & Hall, USA, 1996, 343 str.
4. Bernus P., Mertins K., Schmidt G.: *Handbook on Architectures of Information Systems*, Springer – Verlag, Berlin/Heidelberg, 1998, 834 str.
5. Booch Grady, Rumbaugh James, Jacobson Ivar: *The Unified Modeling Language User Guide*. Reading, Massachusetts: Addison- Wesley, 1998, 482 str.
6. Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie : *Informationsverarbeitung in der Konstruktion '99* : Tagung Berlin, 19. und 20. Oktober 1999, Düsseldorf : VDI, 1999, 506 str.
7. Corbett J., Donner M., Meleka J., Pym C.: *Design for Manufacturing, Strategies, Principles and Techniques*, Addison – Wesley Publishing company, Workingham, England, 1991, 357 str.
8. Downs E., Clare P., Coe I.: *Structured System Analysis and Design Method Application and Context*, Prentice Hall, New York, 1992, 407 str.
9. Duhovnik J., Tavčar J.: *Elektronsko poslovanje in tehnični informacijski sistemi: PDMS – products data management systems*, Ljubljana: LECAD, Fakulteta za strojništvo, 2000, 177 str.
10. Duncan J., Thorpe B., Summer P.: *Quality Assurance in Construction*, Gower, England, 1990, 132 str.
11. Flood R. L.: *Beyond TQM*, John Wiley & Sons, Chichester 1994, 303 str.
12. French M. J.: *Conceptual design for Engineers*, Springer- Verlag, Berlin, Heidelberg, 1985, 226 str.
13. Fripp J.: *Learning through simulations, A guide to the design and use of simulations in business and education*, McGraw- Hill book company, London, 1993, 181 str.
14. Gaus W.: *Dokumentation und Ordnungssystemen*, Springer, Verlag, Berlin, Heidelberg, 1995, 504 str.
15. Grabowski H., Rude S., Grein G.: *Universal Design Theory*, Shaker Verlag, 1998, 369 str.

16. Grabowski H., Lossack R. S., Weißkopf J.: Datenmanagement in der Produktentwicklung: automatische Klassifikation von Produktdaten aus 3D-CAD-Systemen, PDM- und ERP-Systemen, XML- und Office-Dokumenten, ..., München, Wien : Hanser, cop. 2001, 248 str.
17. Graham R. J., Englund R. L.: *Creating an environment for successful project*, Jossey-Bass Inc., San Francisco, California, 1997, 253 str.
18. Hubka V., Eder W. E.: *Theory of Technical Systems*, Springer – Verlag, 1988, 275 str.
19. Jones T.: *New Product Development, An introduction to a multifunctional process*, Butterworth-Heinemann, Oxford, 1997, 134 str.
20. Kalpič B.: *Modeliranje poslovnih procesov*, Fakulteta za strojništvo, Maribor, 1998, 68 str.
21. Koller R., Kastrup N.: *Prinziploesungen zur Konstruktion technischer Produkte*, Springer – Verlag, Berlin 1994, 476 str.
22. Kos M.: *Konstruktivska dokumentacija*, Tehniška založba, Ljubljana, 1985, 261 str.
23. Kurbel K.: *Produktionsplanung und –steuerung: methodische Grundlagen von PPS-Systemen und Erweiterungen*, R. Oldenbourg Verlag, Muenchen, Wien, 1993, 364 str.
24. Lipnack J., Stamps J.: *Reaching Across Space, Time, and Organization with Technology*, John Wiley & Sons, New York, 1997, 262 str.
25. McIntosh K. G.: *Engineering Data Management, A guide to Succesful Implementation*, McGrow – Hill, London, 1995, 279 str.
26. Perlitz M., Offinger A., Reinhardt M., Schug K.: *Reengineering zwischen Anspruch und Wirklichkeit, Ein managementansatz auf dem Pruefstand*, Gabler Verlag, Wiesbaden, Germany, 1996, 360 str.
27. Pidd M.: *Computer simulation in management science*, J. Wiley & Sons, Chichester, 1998, 279 str.
28. Prasad B.: *Concurrent Engineering Fundamentals, Vol. I, Integrated product and process organization*, Technimic, Lancaster, USA, 1996, 478 str.
29. Prasad B.: *Concurrent Engineering Fundamentals, Vol. II, Integrated product and process organization*, Technimic, Lancaster, USA, 1997, 490 str.
30. Prodan N.: *Upravljanje z razvojnimi podatki in dokumentacijo izdelkov skozi njihovo življensko dobo*, Zbornik posvetovanja - Dnevi slovenske informatike 2004, Ljubljana, Slovensko društvo informatika, 2004, str. 388 – 389

31. Produkte entwickeln im realen Umfeld : *was bringen neue Werkzeuge wie 3D-CAD/CAM, EDM/PDM und Virtualisierung?* : Tagung München, 9. und 10. November 2000, Düsseldorf : VDI, 2000, 397 str.
32. Prozessketten für die virtuelle Produktentwicklung in verteilter Umgebung : *Informationsverarbeitung in der Konstruktion '98* : Tagung München, 20. und 21. Oktober 1998, Berlin [etc.] : VDI, 1998, 449 str.
33. Reinertsen D.: *Managing the Design Factory, A Product Develer's Toolkit*, The free press, New York, USA, 1997, 269 str.
34. Rumbaugh J., Jacobson I., Booch G.: *The Unified Modelling Language Reference Manual*: Addison-Wesley, B.I., 1998, 482 str.
35. Scheer A. W.: *Business Process Engineering, Reference Models for Industrial Enterprices*, Springer – Verlag, Heidelberg, Germany, 1994, 770 str.
36. Schichtel M.: *Produktdatenmodellierung in der Praxis*, München, Wien : Hanser, cop. 2002, 288 str.
37. Spinner M. P.: *Project Management, Principles and Practices*, Prentice – Hall, 1997, 306 str.
38. Stark J.: *Engineering Information Management Systems, Beyond CAD/CAM to Concurrent Engineering Support*, Van Nostrand Reinhold, New York, 1992, 209 str.
39. Sturm J.: *VB6 UML Design and Development* : Wrox press Ltd., 1999, 581 str.
40. Turban E., et al: *Information Technology for Management*, Upper Saddle River, New York: Prentice-Hall, 1999, 118 str.
41. Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung : *Potentiale und Erfahrungen : Tagung Stuttgart*, 8. und 9. Juni 1999, Düsseldorf : VDI, 1999, 234 str.
42. Ward J.: *Principles of Information System Management*. London, Routledge, 1995, 267 str.

11.Viri

1. Ed Miller: Strong Worldwide Growth.
[URL: <http://www.memagazine.org/backissues/october98/features/pdm/strong.html>], 1998
2. Bartuli G., Bourke R.: The Best of Both Worlds
[URL: <http://www.pdmic.com/articles/bestbth2.html>], 1995
3. UML Resource Page. [URL: <http://www.omg.org/uml/>], Object Management Group, 2.4.2003.

12. Priloga

Tabela 2: Preverjanje upravičenosti prilagoditve in pošiljanje VP – ITS

PRIMER UPORABE: PREVERJANJE UPRAVIČENOSTI PRILAGODITVE IN POŠILJANJE VP – ITS

Pregled

Namen je potrditev izdelave prilagoditve s strani svetovalca in posredovanje vodji projekta - ITS

Primarni akter

Svetovalec

Sekundarni akter

Ni

Začetna točka

Vnesena zahteva za prilagoditev

Končna točka

Zahtevek sprejet s strani svetovalca, ocenjeni stroški in posredovana vodji projekta - ITS

Izmerljiv rezultat

Zahtevek posredovan VP - ITS

Diagram poteka

Svetovalec vnese oceno stroškov (potrebno število dni). Zahtevi spremeni status, kar povzroči avtomatsko elektronsko sporočilo VP – ITS in avtomatsko vezavo zahtevka v delovno mapo VP – ITS.

Alternativen diagram poteka

Ni

Razširitve primera uporabe

Svetovalec preveri, če obstaja kakšna alternativna rešitev problema. Pomaga si lahko z bazo gotovih in zavrnjenih prilagoditev. Če je edini izhod izvajanje prilagoditve, jo potrdi in pošlje VP – ITS.

Nerešeni problemi

Ni

Tabela 3: Potrditev izdelave prilagoditve

PRIMER UPORABE: POTRDIČEV IZDELAVE PRILAGODITVE

Pregled

Namen primera uporabe je potrditev izvajanja prilagoditve

Primarni akter

VP - ITS

Sekundarni akter

VP – stranka

Začetna točka

VP – ITS ima zahtevek za prilagoditev potrjen s strani svetovalca

Končna točka

Prilagoditev je odobrena ali ne

Izmerljiv rezultat

Odobrena prilagoditev

Diagram poteka

Akter najprej pregleda pogodbo s stranko. Če je izvajanje prilagoditve zajeto v pogodbi, prilagoditev odobri, ji spremeni status in jo s tem pošlje vodji razvijalcev v delo (obenem pošlje tudi avtomatsko elektronsko sporočilo)

Alternativen diagram poteka

V kolikor izvajanje prilagoditve ni zajeto v pogodbi, je to dodatno delo, ki ga mora odobriti še VP – stranka. Torej obstaja še vmesen korak – odobritev stranke.

Razširitve primera uporabe

Ni

Nerešeni problemi

Ni

Tabela 4: Razporeditev izvajanja dela razvijalcu

PRIMER UPORABE: RAZPOREDITEV IZVAJANJA DELA RAZVIJALCU

Pregled

Namen primera uporabe je razporeditev izvajanja prilagoditve razvijalcu

Primarni akter

Vodja razvijalcev

Sekundarni akter

Ni

Začetna točka

Potrjena zahteva za prilagoditev

Končna točka

Izvajanje prilagoditve je razporejeno razvijalcu

Izmerljiv rezultat

Razporejena prilagoditev

Diagram poteka

Akter poišče v bazi dela razvijalcev prostega razvijalca in temu primerno spremeni status zahtevka. S tem je zahtevek poslal v delovno mapo razvijalca in obenem poslal razvijalcu avtomatsko elektronsko sporočilo

Alternativen diagram poteka

Ni

Razširitve primera uporabe

Ni

Nerešeni problemi

Ni

Tabela 5: V testiranje svetovalcu

PRIMER UPORABE: V TESTIRANJE SVETOVALCU

Pregled

Razvijalec po končanem delu pošlje prilagoditev v testiranje svetovalcu

Primarni akter

Razvijalec

Sekundarni akter

Ni

Začetna točka

Prejet zahtevek za izvedbo prilagoditve in končana prilagoditev

Končna točka

Prilagoditev poslana svetovalcu v testiranje

Izmerljiv rezultat

Končana prilagoditev pri svetovalcu

Diagram poteka

Akter po končanem delu vnese v sistem natančen opis novih funkcionalnosti programske opreme – prilagoditev. Zahtevku spremeni status in ga s tem avtomatsko pošlje v svetovalčevo mapo testiranj. Obenem se avtomatsko generira tudi elektronsko obvestilo svetovalcu

Alternativen diagram poteka

Ni

Razširitve primera uporabe

Ni

Nerešeni problemi

Ni

Tabela 6: Potrditev svetovalca in v testiranje stranki

PRIMER UPORABE: POTRDITEV SVETOVALCA IN V TESTIRANJE STRANKI

Pregled

Svetovalec preveri prilagoditev in jo po potrebi pošlje v dodelavo razvijalcu. Ko je zadovoljen, jo potrdi in pošlje v testiranje stranki (ključnemu uporabniku)

Primarni akter

Svetovalec

Sekundarni akter

Ni

Začetna točka

Prejeta prilagoditev od razvijalca

Končna točka

Prilagoditev poslana v testiranje stranki

Izmerljiv rezultat

Končana prilagoditev na testiranju pri stranki

Diagram poteka

Akter po prejemu obvestila preveri funkcionalnost prilagoditve, po potrebi dopolni dokumentacijo in jo pošlje stranki v testiranje.

Alternativen diagram poteka

V kolikor svetovalec ni zadovoljen s prilagoditvijo, ji spremeni ustrezen status in jo pošlje nazaj razvijalcu v dodelavo. Ko ta konča dodelavo, se postopek ponovi.

Razširitve primera uporabe

Ni

Nerešeni problemi

Ni

Tabela 7: Potrditev prilagoditve s strani stranke

<u>PRIMER UPORABE: POTRDIČ PRILAGODITVE S STRANI STRANKE</u>	
<u>Pregled</u>	Stranka (ključni uporabnik) preveri prilagoditev in jo po potrebi pošlje nazaj v dodelavo. Ko je zadovoljen, jo potrdi in pošlje obvestilo svetovalcu.
<u>Primarni akter</u>	Stranka - KU
<u>Sekundarni akter</u>	Ni
<u>Začetna točka</u>	Prejeta prilagoditev od svetovalca
<u>Končna točka</u>	Prilagoditev potrjena
<u>Izmerljiv rezultat</u>	Končana prilagoditev potrjena s strani stranke
<u>Diagram poteka</u>	Akter po prejetju prilagoditve preveri funkcionalnost in jo potrdi. S tem pošlje obvestilo o potrditvi svetovalcu.
<u>Alternativen diagram poteka</u>	V kolikor KU ni zadovoljen s prilagoditvijo, ji spremeni ustrezen status in jo pošlje nazaj v dodelavo. Ko se dodelava konča, se postopek ponovi.
<u>Razširitve primera uporabe</u>	Ni
<u>Nerešeni problemi</u>	Ni

Tabela 8: Razporeditev prilagoditve v bazo gotovih prilagoditev

<u>PRIMER UPORABE: RAZPOREDITEV PRILAGODITVE V BAZO GOTOVIH PRILAGODITEV</u>
<u>Pregled</u> Svetovalec po prejemu potrditve dopolni dokumentacijo o prilagoditvi in jo glede na področje uporabe razporedi v bazo gotovih prilagoditev
<u>Primarni akter</u> Svetovalec
<u>Sekundarni akter</u> Ni
<u>Začetna točka</u> Potrjena prilagoditev s strani stranke
<u>Končna točka</u> Prilagoditev razporejena v bazo gotovih prilagoditev
<u>Izmerljiv rezultat</u> Končana prilagoditev v bazi podatkov
<u>Diagram poteka</u> Akter po prejemu obvestila dopolni dokumentacijo o prilagoditvi. Glede na področje uporabi, ji določi primeren status in jo s tem razvrsti v bazo podatkov gotovih prilagoditev.
<u>Alternativen diagram poteka</u> Ni
<u>Razširitve primera uporabe</u> Ni
<u>Nerešeni problemi</u> Ni

Diagram 4: Diagram zaporedja za potrjevanje izvajanja prilagoditve

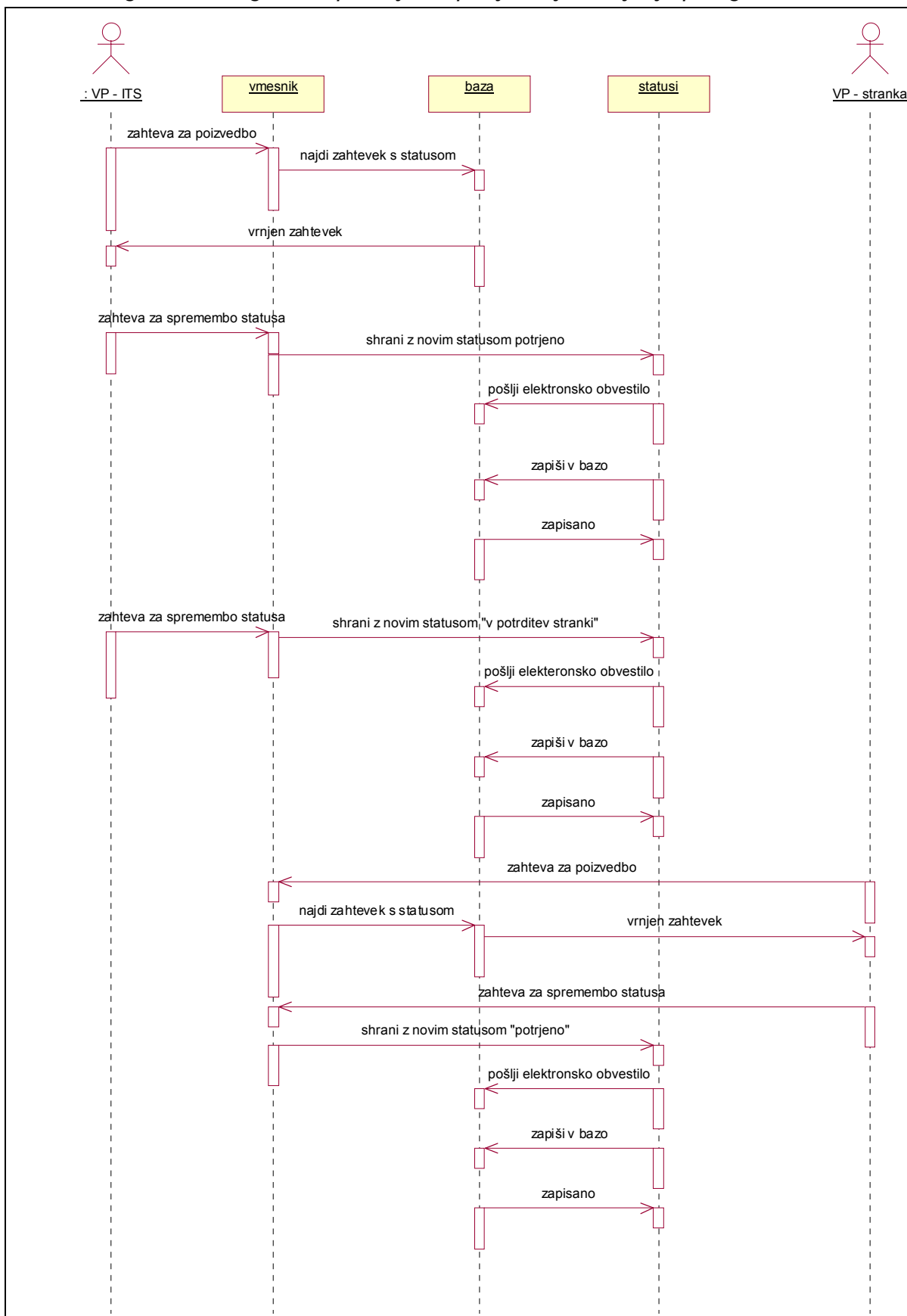


Diagram 4a: Diagram sodelovanja za potrjevanje izvajanja prilagoditve

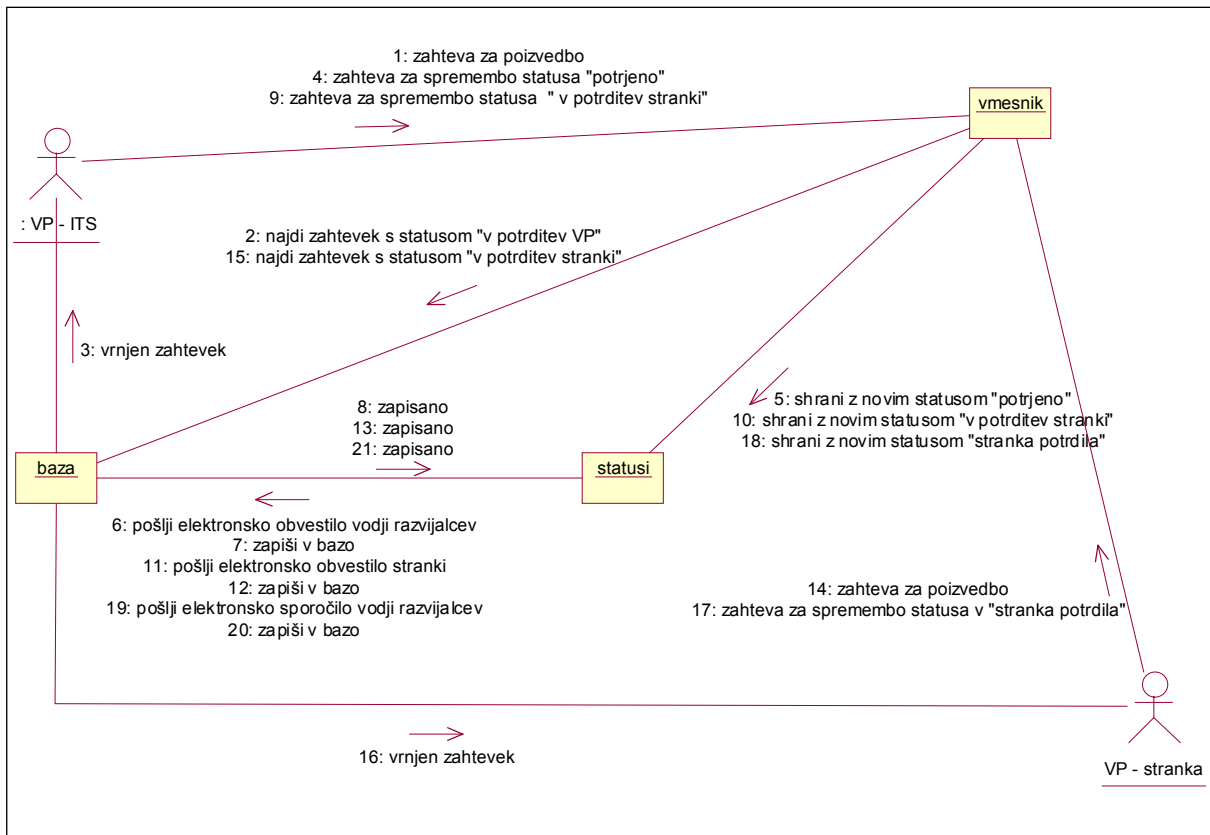


Diagram 5: Diagram zaporedja za razporejanje dela razvijalcu

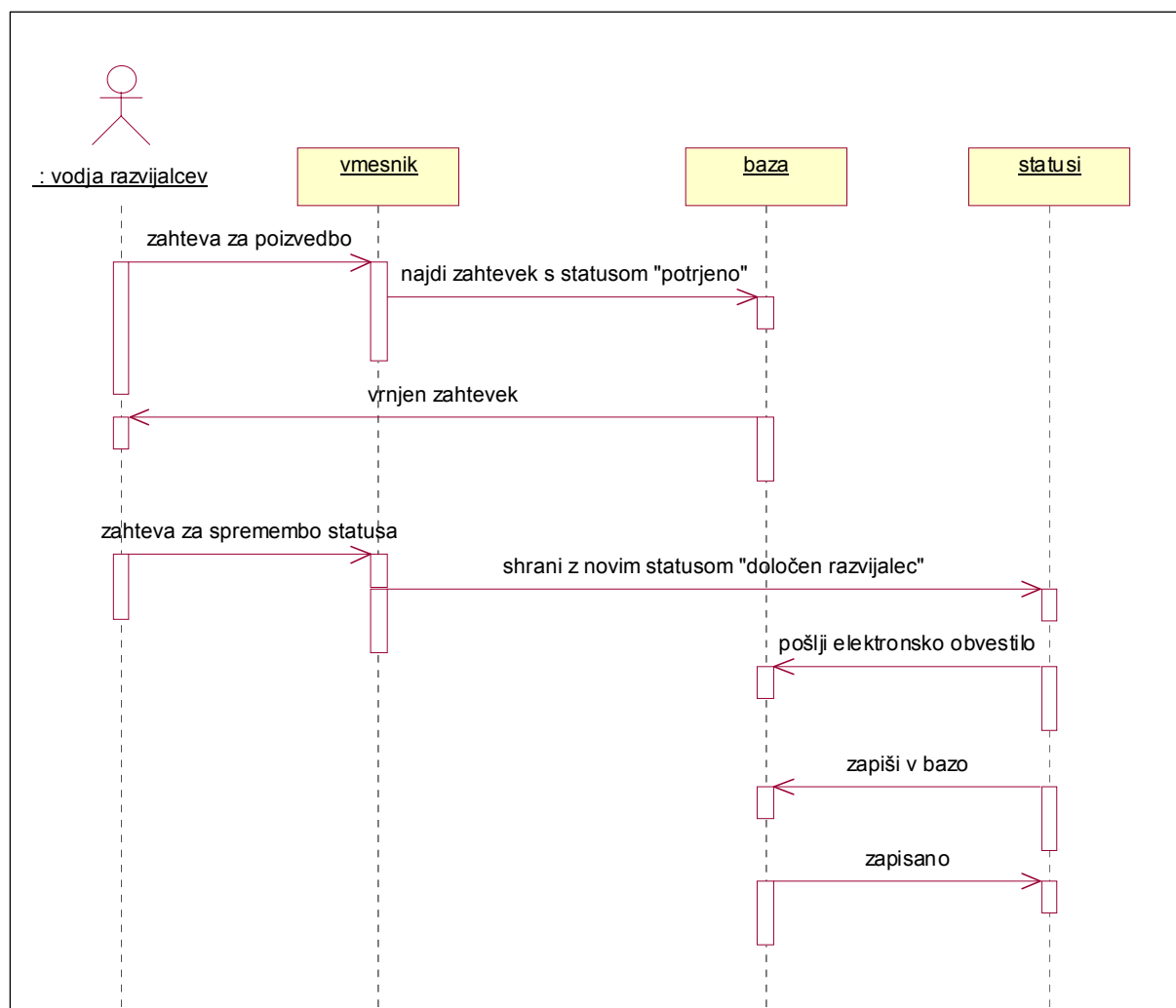


Diagram5a: Diagram sodelovanja za razporejanje dela razvijalcu

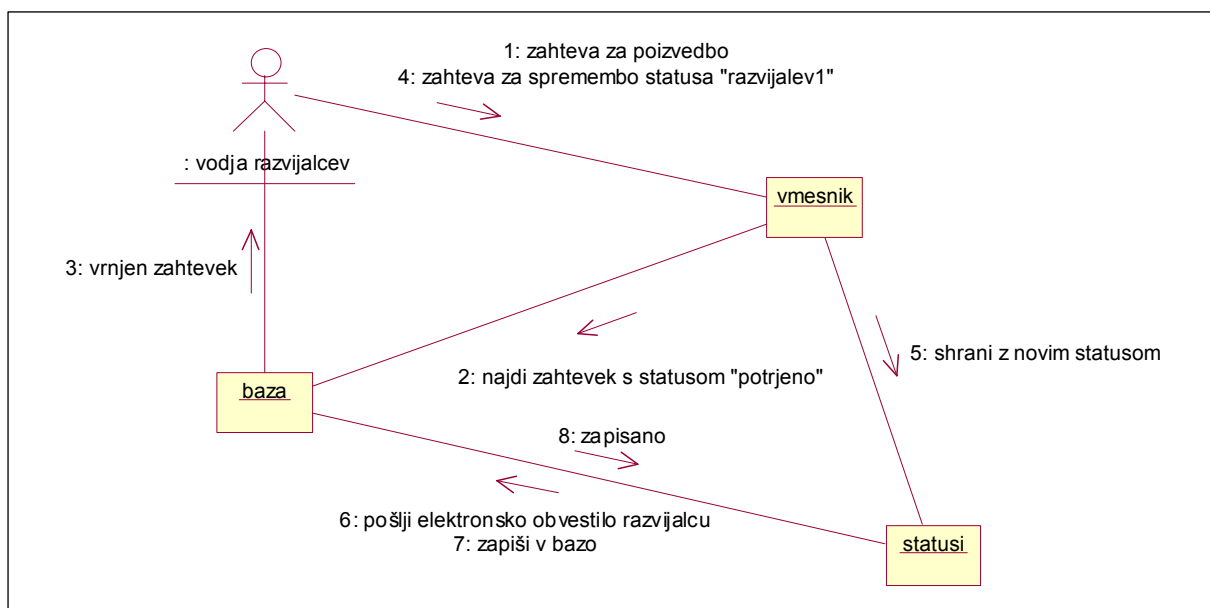


Diagram 6: Diagram zaporedja za pošiljanje v testiranje svetovalcu

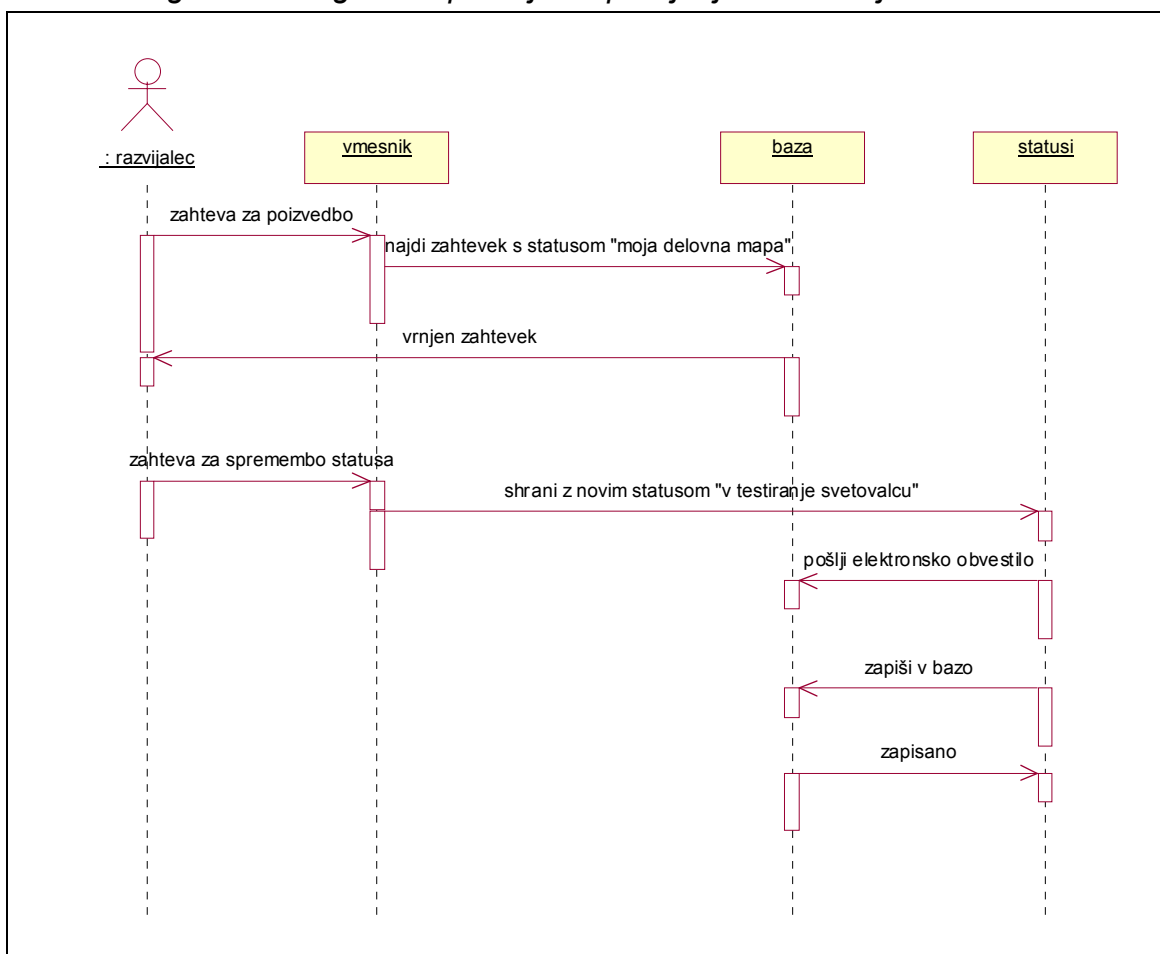


Diagram 6a: Diagram sodelovanja za pošiljanje v testiranje svetovalcu

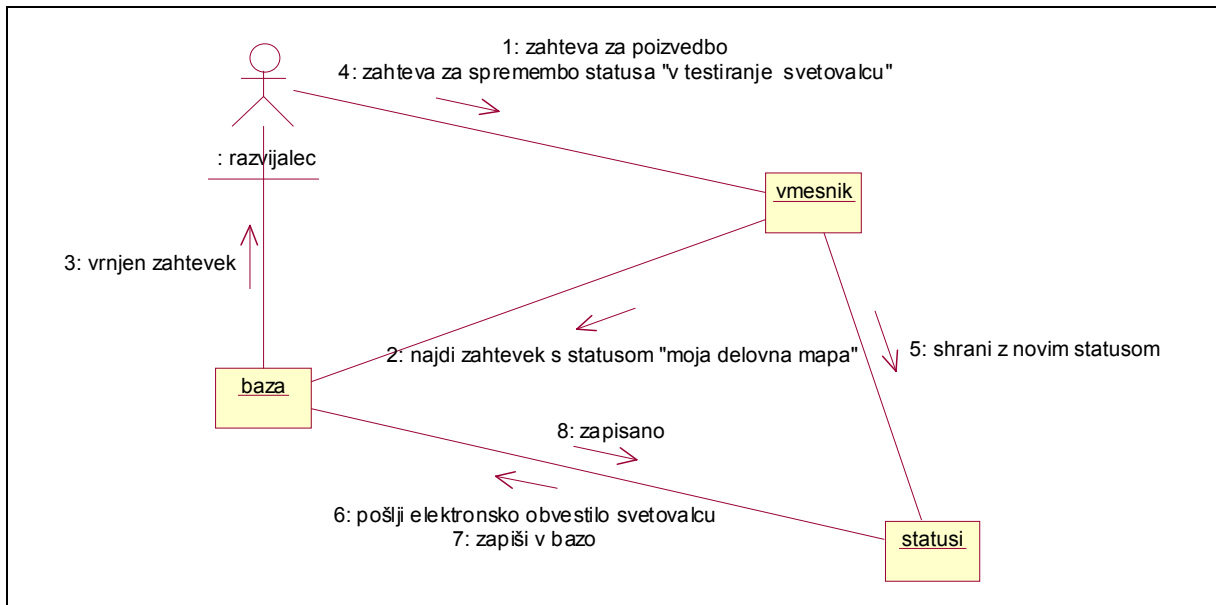


Diagram 7: Diagram zaporedja za pošiljanje v testiranje stranki

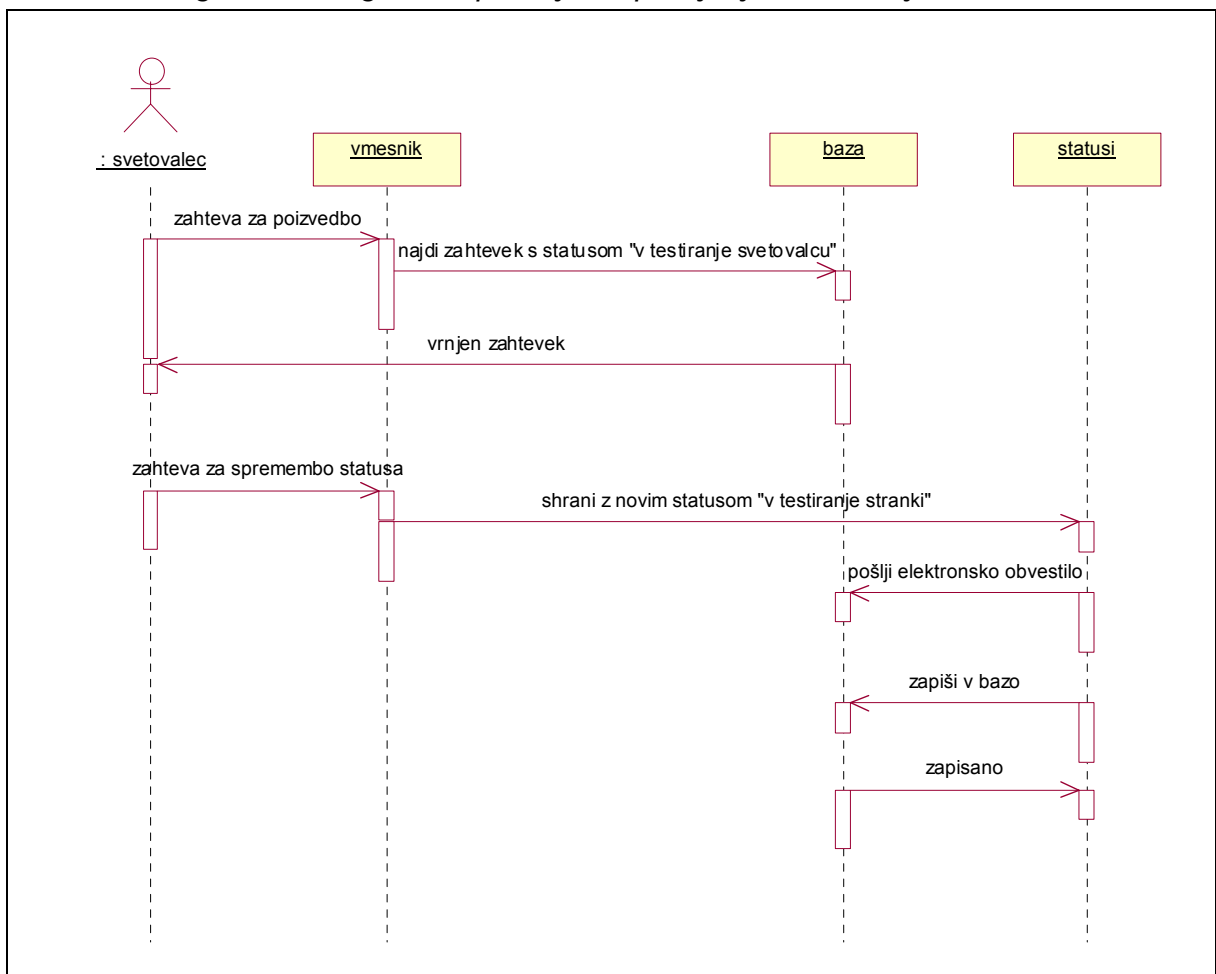


Diagram7a: Diagram sodelovanja za pošiljanje v testiranje stranki

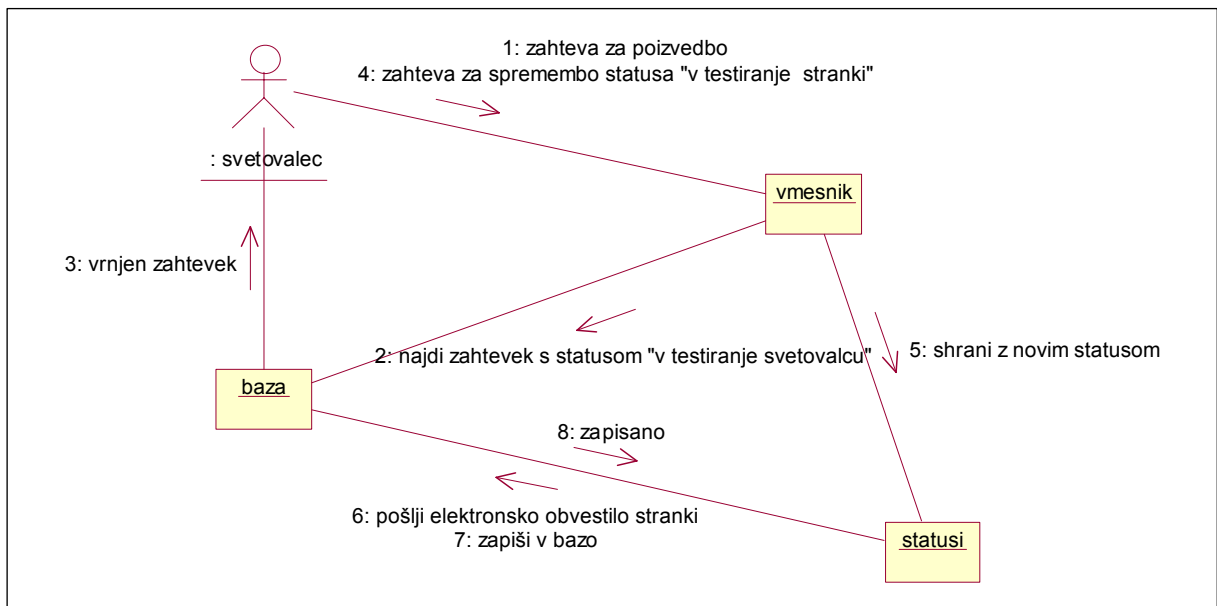


Diagram 8: Diagram zaporedja za testiranje in potrjevanje s strani stranke

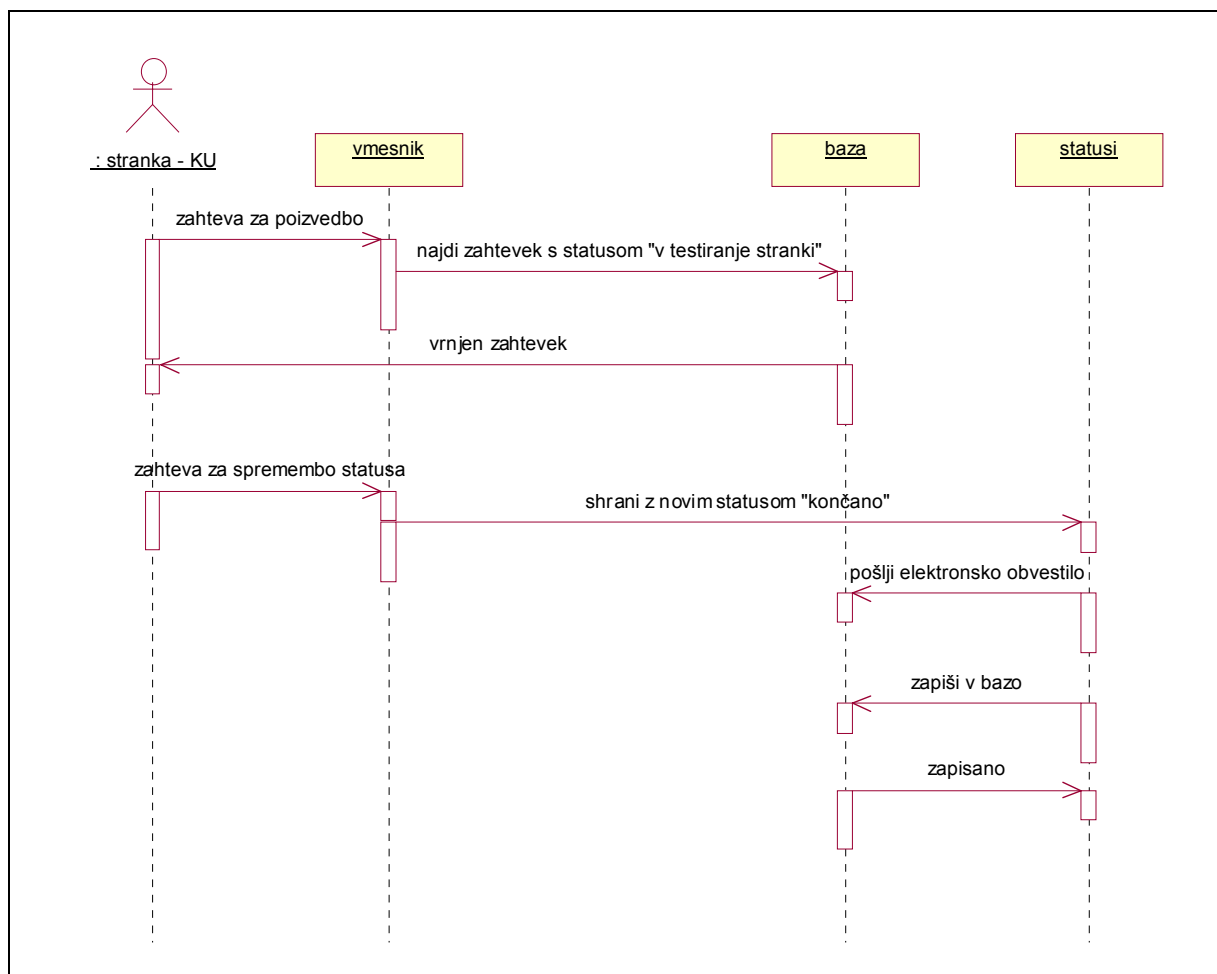


Diagram 8a: Diagram sodelovanja za testiranje in potrjevanje s strani stranke

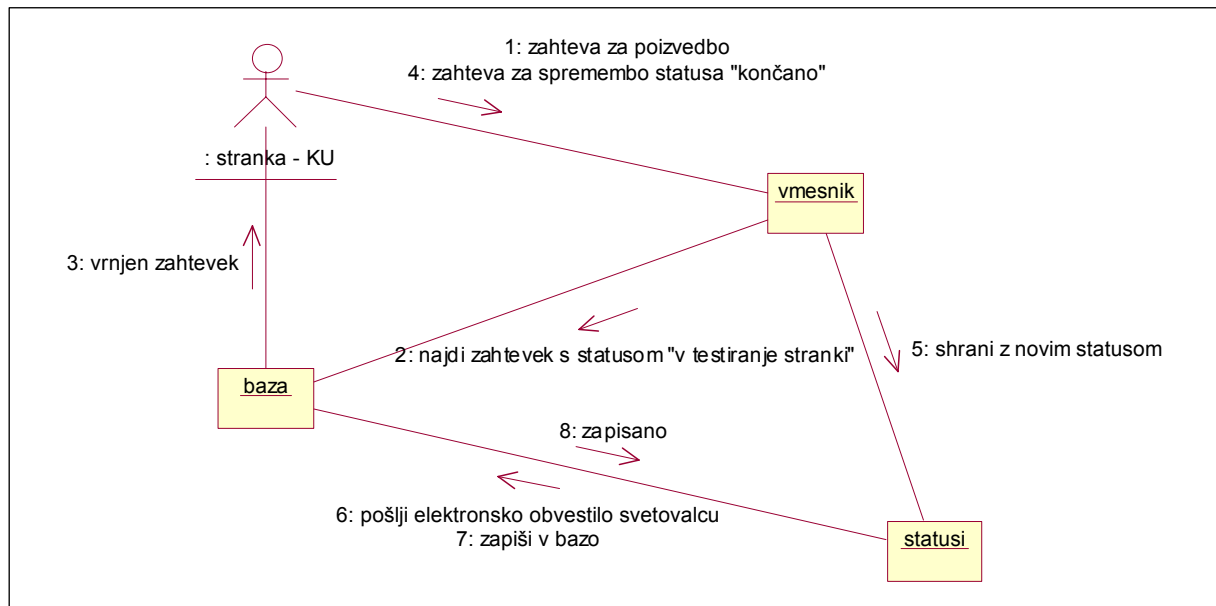


Diagram 9: Diagram zaporedja za razporejanje v bazo gotovih prilagoditev

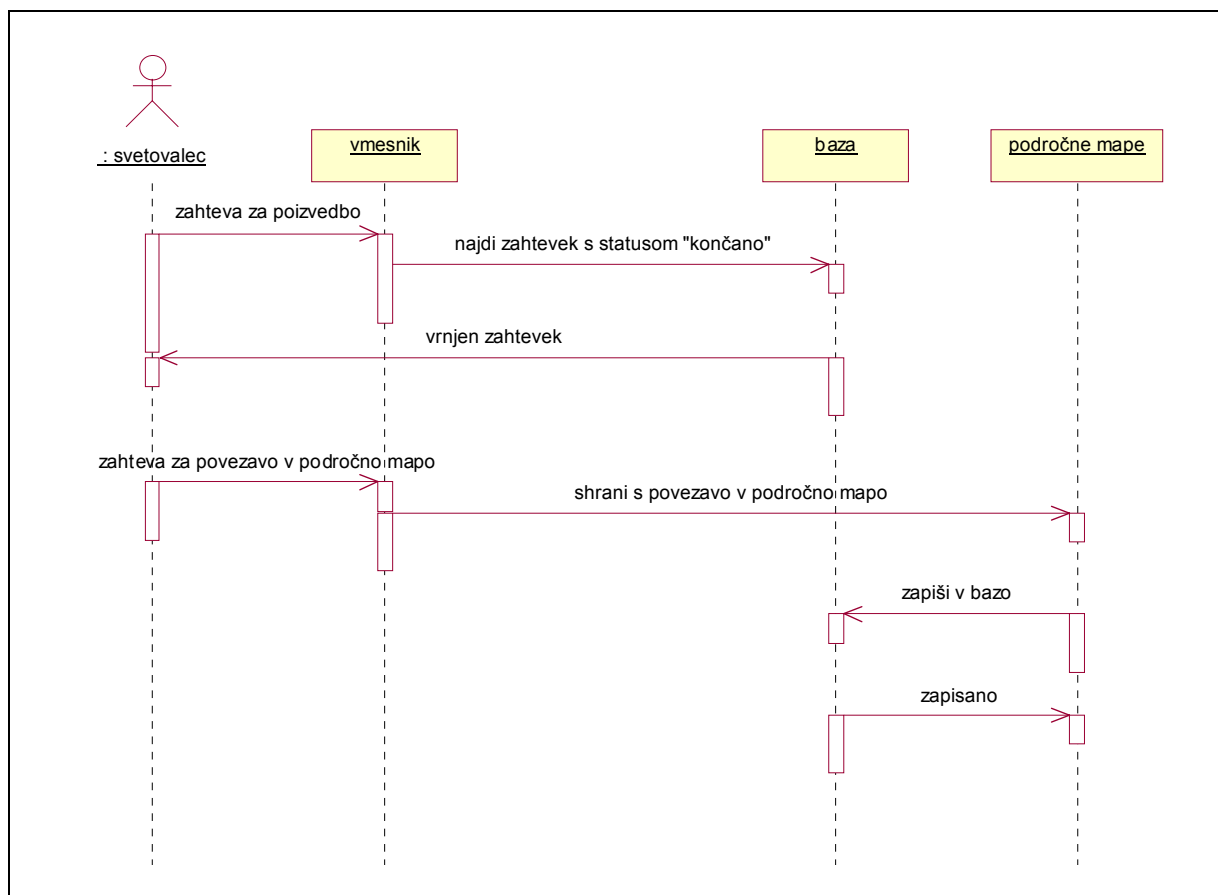


Diagram 9a: Diagram sodelovanja za razporejanje v bazo gotovih prilagoditev

