

Pristop za prilagajanje procesa razvoja programske opreme

Aljaž Zrnec, Damjan Vavpotič

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: aljaz.zrnec@fri.uni-lj.si, damjan.vavpotic@fri.uni-lj.si

Povzetek. V prispevku obravnavamo pristop za prilagajanje procesa razvoja programske opreme potrebam konkretnih projektov, ki se izvajajo v podjetjih za razvoj programske opreme. Predlagan pristop je bistveno preprostejši od dosedanjih pristopov, ki jih opredeljuje področje konstruiranja metodologij. V okviru predstavitve pristopa opredelimo odločitvena pravila, ki usmerjajo postopke izbire ustreznih elementov procesa in same postopke za gradnjo prilagojenih različic procesa, ki je bil v podjetje že prej uveden.

Ključne besede: konstruiranje metodologij, osnovni proces, odločitveno pravilo, karakteristika projekta, prilagajanje procesa

An approach to create project-specific software development process

Extended abstract. In the paper we discuss an approach to software process adaptation to project specific needs. As the proposed approach is much simpler than any known approach suggested by the situational method engineering, it can be easily applied in practice as a special software tool used by method engineers.

The ability of the software process to adapt to project-specific needs is one of the most important factors for assuring software development methodology suitability and acceptance by software engineers in particular organization.

The idea of software development methodology adaptation is not new. Inability of traditional methodologies to adapt to particular situational needs or particular needs of the project, their rigidity and technical unsuitability has led to development of a special scientific field, called situational method engineering (SME). The field of SME is very complex, because it focuses mainly on theoretical aspects of methodology construction and adaptation, and uses rather complex mathematical formalisms for the method construction description and method fragments inquiry from method fragment repository. This is also the main reason why SME has never been widely acknowledged or practiced by software engineers. Also SME also doesn't deal with social suitability and acceptance of the methodology by its users. That has been another reason why the field hasn't been accepted by the engineers.

The key to successful introduction of software development methodologies into organizations, their later use and acceptance, is the ability of its prescribed software process to be adaptable to project-specific needs. The methodology has to define a flexible software process, capable of adapting to specific situations in which new projects emerge.

The aim of this paper is to combine theoretical knowledge from SME and practical knowledge from organizations for software development. On this basis we formally define a new approach for software development process adaptation to projects specific needs. The approach serves as a basis for definition of the decision model which supports software process adaptation.

The proposed approach to process adaptation presented in this paper is based upon selecting the most suitable path

through the given process used in a particular organization. We define the software process used in the observed organization as a *base process*. The approach extends the idea of the SME field theory, of choosing one of several possible paths through a given base process. Our upgrading of the basic idea involves cancellation of all the formerly predefined paths through the process. In our approach, the construction of the adapted instance of the base process is directed by several decision rules. So, for each process element of the base process, the decision rules give an answer about its inclusion in the adapted instance of the base software process.

We use several project characteristics to describe different properties of the projects. Besides technological characteristics, we also define sociological characteristics and customer requirements to be able to describe projects more precisely. Multiple types of characteristics also ensure that in the end the adapted instance of the base process is more precisely attuned to the project at hand.

Keywords: method engineering, base process, decision rule, project characteristics, process adaptation

1 Uvod

Čeprav naj bi uporaba metodologij za razvoj programske opreme pripomogla k njeni boljši kakovosti, izboljšanju razvojnega procesa, standardizaciji poslovanja, lažjemu delu s kadri in ne nazadnje samemu dajanju vtisa naročniku, da se razvoj programske opreme odvija na sistematičen in urejen način, je uporaba metodologij še vedno na dokaj nizki ravni. Raziskave [1] kažejo, da kljub številnim prednostim uporabe metodologij kar 60 odstotkov podjetij za razvoj programske opreme pri izdelavi programskih proizvodov ne uporablja nobene od njih. Samo šest odstotkov podjetij pri svojem delu dosledno sledi predpisani metodologiji, preostali delež podjetij pa ne uporablja nobene metodologije oz. sledi neki neformalni metodologiji, ki se nahaja v obliki skritega oziroma neformaliziranega znanja v glavah razvijalcev.

Zanimivo je tudi, da podjetja, ki metodologije sicer uporabljajo, v njihovi uporabi ne vidijo pomembnejšega prispevka k uspešnejšemu razvoju programske opreme.

Omenjeni problem se še stopnjuje s čedalje večjo kompleksnostjo in obsežnostjo podporne programske opreme. Vsaka generacija programske opreme namreč prinese nove možnosti za uporabo in razširi njen obstoječi nabor funkcionalnosti, kar vodi do obsežnejših sistemov, ki jih je težje načrtovati, izdelati in vzdrževati. Zaradi velikega števila tehničnih novosti in inovacij, ki so na voljo (tri nivojska arhitektura, porazdeljeni sistemi, modelno usmerjeno načrtovanje, elektronsko poslovanje itd.), novi tehnološki vidiki povzročajo preoblikovanje praks pri razvoju programske opreme. Tako se je na tem področju zakoreninilo splošno spoznanje, da razvojne metodologije ne zadovoljujejo potreb organizacijskih sistemov, ne glede na to ali imamo v mislih tehnični, ekonomski ali socialni vidik uporabe metodologij.

Kot odgovor na nastale razmere je že v prvi polovici 90. let nastalo ti. področje konstruiranja metodologij (v nadaljevanju KM), ki se usmerja predvsem na odpravljanje težav, ki jih povzročajo slaba zasnova, neprilagodljivost, togost in tehnična neustreznost tradicionalnih metodologij [2, 3]. Žal pa kljub številnim raziskavam na tem področju lahko ugotovimo le, da se oblike konstruiranja metodologij v praksi niso uveljavile in ostajajo večinoma v domeni akademskih krogov. Razloge za to lahko iščemo predvsem v veliki kompleksnosti obstoječih postopkov za konstruiranje in prilagajanje metodologij, ki večinoma temeljijo preveč na zapletenih matematičnih formalizmih in jih razvijalci oz. metodologi preprosto niso pripravljene sprejeti. Zato v prispevku predstavljamo nov pristop za prilagajanje metodologij [5], ki prilagajanje metodologije bistveno poenostavi. V okviru opredelitve pristopa se usmerimo na razvojni proces, ki je jedro vsake metodologije za razvoj programske opreme.

V okviru drugega razdelka najprej predstavimo oblike KM, nato (razdelek 3) sledi opis formalizacije osnovnega procesa, v razdelku 4 predstavitev pristopa, odločitvenih pravil in postopkov za prilagajanje procesa in v 5. razdelku sklep.

2 Konstruiranje metodologij

KM je inženirska disciplina, ki se ukvarja z načrtovanjem, konstruiranjem in prilagajanjem metodologij, tehnik in orodij za razvoj programske opreme. Posebna veja je situacijsko konstruiranje metodologij (v nadaljevanju SKM), ki se ukvarja s prilagajanjem metodologij potrebam konkretnih projektov oz. konkretnih situacij. Glede na stopnjo prilagodljivosti lahko opredelimo več oblik [4] SKM:

- uporaba togih metodologij: zahteva uporabo metodologij v predpisani obliki (ne gre za pravo obliko SKM).

- izbira med togimi metodologijami: namesto uporabe samo ene toge metodologije lahko izbiramo med več različnimi metodologijami.
- izbira različnih poti v okviru metodologije: omogoča izbor samo tistih delov metodologije (vnaprej določene poti), ki ustreza potrebam določenega projekta (npr.: na voljo imamo razvojni poti za klasičen in hiter razvoj aplikacij).
- izbira in prilagajanje delov metodologije: omogoča izbor posameznih delov iz referenčnih metodologij in njihovo prilagajanje.
- modularno konstruiranje metodologij: omogoča najvišjo stopnjo prilagodljivosti, pri čemer se metodologija zgradi na podlagi vnaprej definiranih gradnikov metodologije, ki so shranjeni v repozitoriju. Gradniki (fragmenti metodologije) se med seboj povezujejo z uporabo pravil v skladne in učinkovite metodologije, ki najbolj ustrezajo potrebam projektov.

Ključni problem, da se izsledki SKM do danes niso uveljavili v praksi, je kompleksnost postopkov konstruiranja, ki temeljijo na zapletenih matematičnih formalizmih, zahtevana visoka raven poznavanja prednosti in slabosti različnih metodologij ter obvladovanje zahtevnih postopkov za iskanje delov metodologij v repozitoriju in njihovo sestavljanje v celoto. Večina pristopov SKM obravnava proces konstrukcije oz. prilagajanja predvsem na ravni metamodeliranja, kar je z akademskega vidika sicer popolnoma ustrezno, v praksi pa zelo slabo sprejeto.

3 Formalizacija osnovnega procesa

3.1 Predpostavke

Pristop za prilagajanje smo zasnovali na naslednjih predpostavkah oz. robnih pogojih:

- prilagaja se proces razvoja: poudariti želimo, da smo osredotočeni na prilagajanje samega procesa, ki je opredeljen z razvojno metodologijo. Poleg tega pa nas zanima izključno postopek prilagajanja in ne konstruiranje procesa z vidika sestavljanja procesa iz njegovih osnovnih gradnikov.
- prilagaja se proces eni organizaciji: predlagani pristop za prilagajanje temelji na uporabi množice odločitvenih pravil, ki je specifična za vsako organizacijo posebej.
- osnova za prilagajanje je vnaprej opredeljena: pristop temelji na prilagajanju procesa, ki je v organizacijo že vpeljan in formaliziran - osnovni proces.
- prilagajanje je omejeno na razvoj poslovnih aplikacij: poudariti želimo, da se ukvarjamo

zgolj s prilagajanjem procesa razvoja programske opreme, čeprav bi teoretično lahko obravnavali tudi širšo skupino procesov, ki niso neposredno povezani z razvojem aplikacij.

- prilagajanje se odvija na ravni posameznega elementa procesa: z omejitvijo želimo preprečiti preveliko razdrobljenost procesa. Elementi procesa se opredelijo v okviru metamodela osnovnega procesa.

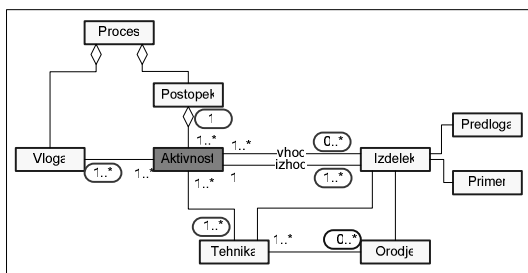
Z navedenimi predpostavkami želimo zagotoviti, da problem prilagajanja procesa programske opreme ne postane preveč kompleksen ali celo neobvladljiv.

3.2 Formalizacija

Kot je bilo omenjeno že v prejšnjem razdelku (3. alineja), predlagani pristop temelji na prilagajanju metodologije in s tem na prilagajanju razvojnega procesa, ki je v organizacijo že vpeljan. Ta proces opredelimo kot osnovni proces, ki je definiran kot:

Def.: Osnovni proces organizacije zajema vse elemente procesa, ki jih opredeljuje metamodel zajetega procesa.

Razvojni proces, ki se izvaja v podjetju, je treba pred izvajanjem samega postopka prilagajanja predhodno formalizirati, tj. ob zajemu na formalni način opisati vse njegove sestavne dele. V okviru formalizacije procesa se izdelata metamodel procesa (Slika 1), ki opredeljuje vse njegove elemente (aktivnost, vloga, izdelek, postopek itd.) in procesni model, ki opredeljuje aktivnosti, relacije med njimi, izdelke in vloge, ki sodelujejo v okviru zajetega osnovnega procesa itd. Vse elemente je treba tudi opisati do določene stopnje podrobnosti.



Slika 1: Primer metamodela zajetega procesa

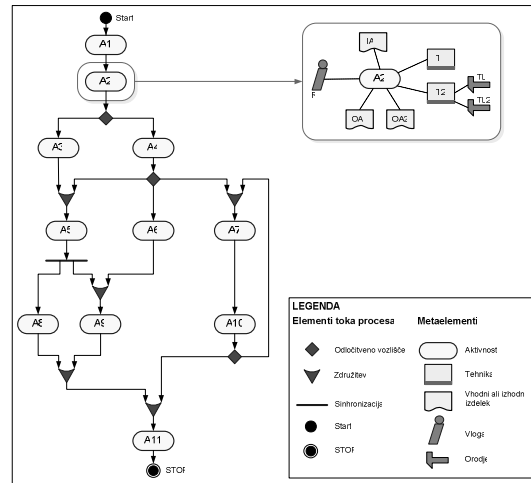
Figure 1: Example of the process metamodel

4 Prilagajanje procesa

4.1 Ideja predlaganega pristopa

Predlagani pristop za prilagajanje procesa individualnim potrebam projektov temelji na iskanju najprimernejše poti skozi model osnovnega procesa (slika 2). Model je predstavljen z diagramom, ki prikazuje aktivnosti, elemente

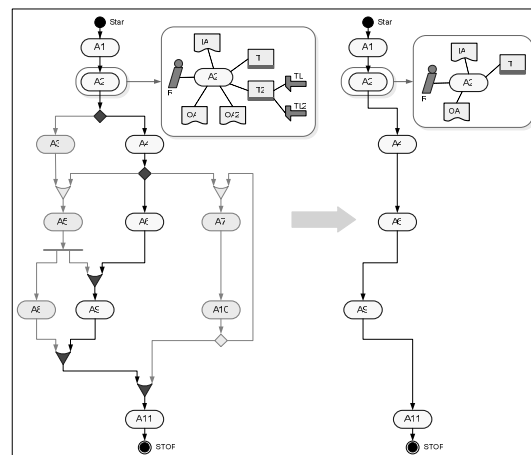
procesnega toka in druge elemente procesa (orodje, vloga, izdelki), ki se vežejo na posamične aktivnosti.



Slika 2: Primer modela osnovnega procesa

Figure 2: Example of the basic process model

Najdena pot predstavlja elemente (aktivnosti, vloge, izdelke itd.), ki sestavljajo prilagojeno različico procesa, ki jo dani projekt nato lahko uporabi (Slika).



Slika 3: Pristop k prilagajanju procesa

Figure 3: Approach to process adaptation

Gradnja prilagojene različice procesa poteka s pomočjo odločitvenih pravil in treh tipov karakteristik (tehnološke, sociološke karakteristike in zahteve naročnika), s katerimi se opišejo lastnosti projekta, za katerega se proces prilagaja. Prikazani pristop temelji na strategiji izbire različnih poti v okviru metodologije (razdelek 2, 3. alineja). Pri tem nadgrajuje omenjeno obliko SKM tako, da odpravi omejitev glede števila mogočih prehodov skozi proces. Predlagani pristop namreč ne predpiše vseh mogočih poti skozi proces vnaprej, ampak poteka prilagajanje procesa tako, da se v različico procesa v odvisnosti od odločitvenih pravil

vstavlja elementi iz osnovnega procesa - postopoma se zgradi pot skozi osnovni proces. Tako je prehodov skozi osnovni proces lahko bistveno več, kot pri prej omenjeni obliki SKM, hkrati pa je pridobljena različica procesa natančneje prilagojena (se bolj prilega) potrebam trenutnega projekta.

4.2 Odločitvena pravila

Zasnovana odločitvena pravila, njihove komponente in povezave med njimi in druge elemente, na katere se navezujejo odločitvena pravila, prikažemo z uporabo metamodela odločitvenih pravil (Slika 4). Splošno omejitveno pravilo (osrednji element) v metamodelu pomeni posplošitev štirih skupin pravil: pravil procesnega toka, strukturnih pravil, pravil celovitosti in pravil konsistentnosti. Poleg teh smo opredelili še dve skupini pravil, in sicer osnovna in izpeljana dejstva.

Pravilo procesnega toka ali strukturno pravilo določa obstoj določene povezave v modelu osnovnega procesa in ima naslednjo obliko:

IF x AND pogoj THEN y,

kjer x in y pomenita začetni in končni element povezave, na katero je vezano pravilo, pogoj pa vrednosti karakteristik projekta, za katere bo IF del stavka (premisa) resničen. Pogoj ima naslednjo obliko:

EP1 lo EP2 lo...lo EPn,

kjer so EP1...EPn elementi pogoja, ki pomenijo vrednosti karakteristik projekta lo pa logični operatorji. Primer takega pravila bi tako lahko bil:

IF "Odločitveno vozlišče 1" AND obseg sistema = velik AND življenski cikel = inkrementalni THEN "Analiziraj logično strukturo sistema".

Pravila celovitosti se nanašajo na metamodel osnovnega procesa. Z njihovo pomočjo je mogoče preveriti, ali prilagojena instanca procesa vključuje vse zahtevane elemente procesa. Pravilo celovitosti ima naslednjo obliko:

IF metaelement x ADN metaelement y AND pogoj THEN akcija,

kjer metaelement x in y pomenita začetni in končni element metapovezave, na katero se pravilo nanaša v metamodelu, pogoj je števnost metapovezave (obkrožene števnosti na sliki 1), akcija pa, kaj se mora zgoditi, če komponenta pogoj v premisi pravila ni resničen. Pogoj v pravilu ima namreč naslednjo obliko:

NOT (min =< s =< max),

s katerim se opredeli najmanjše in največje število elementov tipa metaelement y, s katerimi se lahko povezuje tip elementa, ki je opredeljen v okviru komponente metaelement x.

Pravila konsistentnosti zagotavljajo, da je izbor elementov procesa v okviru prilagojene različice procesa konsistenten. Pravilo konsistentnosti za element x namreč opredeli množico elementov D, ki morajo biti vključeni v prilagojeno različico procesa, če se vanjo vključi element procesa x.

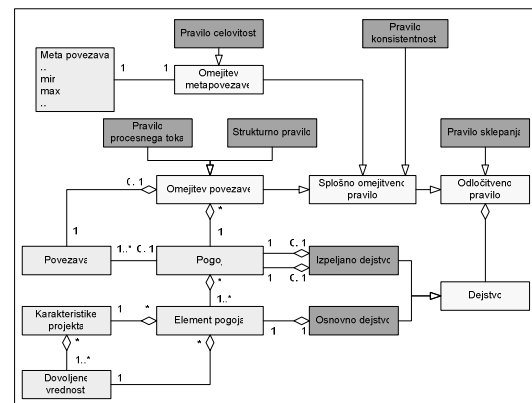
Množica osnovnih dejstev pomeni vhodne podatke o projektu. Na podlagi ujemanja osnovnih dejstev z elementi pogoja (EP1...EPn) v premisi odločitvenega pravila se za vsako pravilo ugotovi, ali je vrednost pogoja resnična ali neresnična. Osnovno dejstvo ima naslednjo obliko:

KPx PO "vrednost x",

kjer PKx označuje karakteristiko projekta, PO je primerjalni operator, "vrednost x" pa pomeni dejansko vrednost karakteristike za obravnavani projekt. Ujemanje z osnovnimi dejstvi poteka v okviru treh skupin pravil: pravil procesnega toka, strukturnih pravil in pravil sklepanja.

Izpeljano dejstvo se pridobi iz osnovnih dejstev s pomočjo pravil sklepanja. Izpeljano dejstvo je lahko sestavljeno iz več elementov, ki imajo enako obliko kot elementi pogoja v okviru pravil procesnega toka. Tako za osnovna kot za izpeljana dejstva je značilno, da se pridobivajo dinamično, hkrati z izvajanjem postopkov gradnje prilagojene različice procesa [6].

V okviru predlaganega pristopa smo predvideli tudi uporabo sistema za izvajanje pravil, ki uporablja pravila sklepanja. Pravila sklepanja se opredelijo z namenom predstavitve relacij odvisnosti med karakteristikami projektov. Mehanizem sklepanja tako na podlagi premis v pravilih sklepanja izpelje določene sklepe, ki so predstavljeni z novimi (izpeljanimi) dejstvi. Izpeljana dejstva se nato uporabijo v pogojih strukturnih pravil in pravil procesnega toka med gradnjo prilagojene različice procesa.



Slika 4: Metamodel odločitvenih pravil

Figure 4: Decision rules metamodel

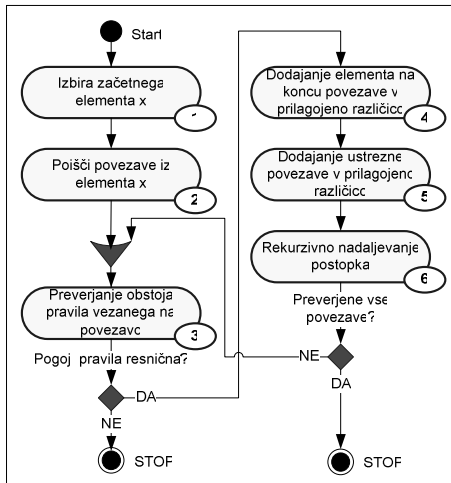
4.3 Postopki predlaganega pristopa

Pristop za prilagajanje procesa predvideva uporabo štirih postopkov. Izvedba prvih treh naštetih postopkov je obvezna, izvedba zadnjega pa opcijska. To so:

- postopek za prilagajanje,
- postopek za preverjanje celovitosti,
- postopek za preverjanje konsistentnosti in
- postopek za pojasnjevanje rezultata.

4.3.1 Postopek prilaganja

Postopek prilaganja je temeljni postopek v okviru predlaganega pristopa. Postopek na podlagi elementov osnovnega procesa zgradi različico procesa za obravnavani projekt. Pri tem upošteva odločitvena pravila in karakteristike konkretnega projekta (Slika).



Slika 5: Postopek prilaganja

Figure 5: Adaptation process

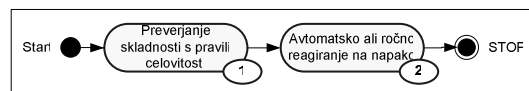
V okviru prve aktivnosti postopka se iz modela osnovnega procesa izbere začetni element procesa (aktivnost 1) prilagojene različice. V naslednji aktivnosti (aktivnost 2) se poiščejo povezave obravnavanega elementa s sosednjimi elementi. Te povezave tvorijo množico povezav L . Sledi zanka, v kateri se za vsako povezavo iz množice L ugotovi, ali je nanjo vezano kakšno odločitveno pravilo (aktivnost 3). Če je povezavi pripisano odločitveno pravilo (strukturno pravilo ali pravilo procesnega toka) in je pogoj v njegovi premisi resničen, se vozlišče iz modela osnovnega procesa, ki se nahaja na koncu omenjene povezave, doda v model prilagojene različice (aktivnost 4). V okviru četrte aktivnosti lahko nastopita tudi naslednja dva dogodka:

- logične vrednosti pogoja v premisi pravila ni mogoče izračunati. V tem primeru se postopek ustavi in zahteva posredovanje metodologa.
- če povezavi (usmerjeni ali neusmerjeni) ni pripisano nobeno pravilo, to pomeni, da se mora element na njenem koncu obvezno nahajati v prilagojeni različici.

V model prilagojene različice se zatem doda tudi obravnavana povezava (aktivnost 5), kateri je lahko pripisano odločitveno pravilo ali pa ne. V nadaljevanju postopka sledi rekurzivni klic taistega postopka, katerega parameter je element, ki je bil dodan v graf prilagojenega procesa (aktivnost 6). Ko postopek preveri vse povezave, ki vodijo iz trenutnega vozlišča, se ta konča.

4.3.2 Postopek za preverjanje celovitosti in konsistentnosti

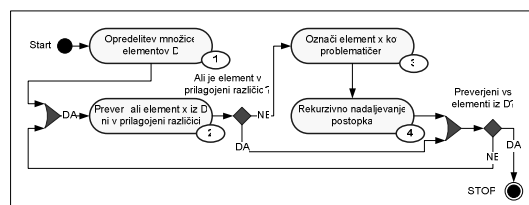
V okviru postopka za preverjanje celovitosti (Slika) pristop zagotovi celovitost prilagojene različice procesa. To se izvede z uporabo pravil celovitosti (aktivnost 1), ki na podlagi omejitev, opredeljenih na ravni metamodela osnovnega procesa, v aktivnosti 2 sprožijo ustrezne akcije (glej razdelek 4.2).



Slika 6: Postopek za preverjanje celovitosti

Figure 6: Completeness check

V okviru postopka za preverjanje konsistentnosti (Slika) se za vsak element, ki se vključi v prilagojeno različico procesa, preveri, ali zanj obstaja množica elementov iz osnovne metodologije, od katerih je odvisen. Če katerega od elementov iz najdene množice ni v prilagojeni različici, se element, za katerega se izvaja primerjanje, označi kot problematičen. Problematičen element mora obravnavati metodolog, ki sprejme odločitev o tem, ali bo element vključen v različico procesa ali ne.

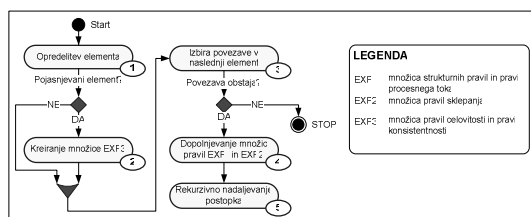


Slika 7: Preverjanje konsistentnosti

Figure 7: Consistency check

4.3.3 Postopek za pojasnjevanje rezultata

Obravnavani postopki v okviru predlaganega pristopa opredeljujejo celovit odločitveni model, ki podpira gradnjo prilagojenih različic procesa. Ker prilagojena različica procesa nastane na podlagi odločitev, do katerih pridemo z uporabo odločitvenih pravil, in ker vsak odločitveni model omogoča pridobiti odgovor, zakaj je bila določena odločitev sprejeta, si želimo, da je tudi za vsak element procesa v prilagojeni različici mogoče pojasniti, na podlagi katerih pravil je bil vključen v različico. To omogoča postopek za pojasnjevanje rezultata (Slika).



Slika 8: Postopek pojasnjevanja rezultata

Figure 8: Explanation of the result

Pojasnilo je predstavljeno s potjo od pojasnjevanega elementa do startnega elementa v modelu prilagojene različice. Pri tem se tvorijo množice odločitvenih pravil EXP1, EXP2 in EXP3 (glej legendo na sliki 7), ki pojasnjujejo vključenost pojasnjevanega elementa v različico procesa.

5 Sklep

Prednost predlaganega pristopa je poleg relativne preprostosti tudi to, da so pridobljene različice procesa prilagojene tudi s sociološkega vidika, saj so sestavljene iz elementov osnovnega procesa, ki so jih razvijalci v organizaciji že sprejeli, saj jih že uporabljajo.

Na podlagi predstavljenega pristopa smo tudi že izdelali programski modul, ki podpira prilagajanje razvojnega procesa. Omenjeni modul je del obširnejšega orodja, ki podpira celoten razvojni cikel metodologije, od uvedbe v organizacijo, do njenega prilagajanja in poznejšega izboljševanja. Ker modul za prilagajanje procesa zaenkrat še ni povezan z ustreznim programskim modulom, ki podpira izvajanje sklepanja, se bomo v okviru nadaljnjega dela osredotočili predvsem na izbiro ustrezne metode sklepanja (podatkovno usmerjene ali ciljno usmerjene) in s tem tudi na izbor ustreznega programskega modula, ki bo to metodo sklepanja podpiral in omogočil učinkovito integracijo v celotno prej omenjeno programsko orodje.

6 Literatura

- [1] FITZGERALD, B. (1998). An empirical investigation into the adoption of system development methodologies. *Information & Management* 34, pp. 317-328
- [2] BRINKKEMPER, S., M. SAEKI and F. HARMSSEN (1999). Meta-modeling based assembly techniques for situational method engineering. *Information Systems*, Vol. 24, No. 3, pp. 209-228
- [3] HENDERSON-SELLERS, B. (2003). Method engineering for OO systems development. *Communications of the ACM*, Vol. 46, No. 10, pp. 73-78
- [4] HARMSSEN, F., S. BRINKKEMPER, H. OEI (1994). Situational Method Engineering for information system project approaches, Methods and associated tools for the information systems life cycle, Amsterdam, 169-194

[5] ZRNEC, A. (2006). *Odločitveni model za prilagajanje procesa razvoja informacijskih sistemov individualnim potrebam projektov*: doktorska disertacija. Ljubljana, 2006, 183 str.

[6] BAJEC, M., D. VAVPOTIČ, M. KRISPER (2007). Practice-driven approach for creating project-specific software development. *Information and software technology*, Vol. 49, pp. 345-365

Aljaž Zrnc je magistriral leta 2002 na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Leta 2006 je doktoriral s področja konstruiranja metodologij. Zaposlen je v Laboratoriju za informatiko kot asistent za področje podatkovnih baz. Na raziskovalnem področju se ukvarja s strukturnim razvojem IS, konstruiranjem metodologij in podatkovnimi zbirkami. Je avtor ali soavtor številnih prispevkov v strokovnih in znanstvenih publikacijah.

Damjan Vavpotič je magistriral in doktoriral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Zaposlen je v Laboratoriju za informatiko na Fakulteti za računalništvo in informatiko kot asistent za področje računalništva in informatike. Njegova interesna področja med drugimi obsegajo metodologije razvoja informacijskih sistemov, objektni pristop k razvoju programske opreme in razvoj programske opreme na podlagi modelov. Je avtor ali soavtor številnih prispevkov v strokovnih in znanstvenih publikacijah.