

# ■ Pristop in programska podpora za prilagajanje procesa razvoja informacijskega sistema

Aljaž Zrnec, Marko Bajec, Marjan Krisper

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška 25, 1000 Ljubljana  
{aljaz.zrnec, marko.bajec, marjan.krisper}@fri.uni-lj.si

## Povzetek

Prispevek predstavi opravljeno delo v okviru razvoja aplikacije AMT, pri čemer se osredotočimo na opis modula za prilagajanje procesa razvoja informacijskega sistema konkretnim potrebam projektov – MethAdapt. V okviru tega se tematika usmeri na predstavitev konceptualne zgradbe odločitvenega modela za prilagajanje procesa in na predlagani pristop za prilagajanje procesa, ki temelji na uporabi več vrst odločitvenih pravil. V prispevku je predstavljen postopek prilagajanja in uporaba odločitvenih pravil v okviru tega postopka.

**Ključne besede:** center odličnosti, prilagajanje procesa, programsko orodje, konstruiranje metodologij, graf, karakteristika

## Abstract

**The approach and software tool for adapting the is development process**

In the paper we present our work in the area of developing the software tool amt. We focus on a particular software module called methadapt, used for software process adaptation to specific needs of a project. We present conceptual decision model and the approach behind it, used for process adaptation. The approach is based on different kinds of decision rules which direct the construction of process instances. In the paper our own approach to process adaptation and the usage of decision rules is explained.

**Keywords:** center of excellence, adapting the process, software tool, construction of methodology, graph, characteristics

## 1 Uvod

Tematika prispevka predstavlja del raziskovalnega dela, ki smo ga opravili v okviru izdelave doktorske disertacije, ki je nastala na fakulteti za računalništvo in informatiko [1]. V okviru projekta Centra odličnosti – Informacijske in komunikacijske tehnologije in storitve (v nadaljevanju CO ICT) poteka v laboratoriju za informatiko Fakultete za računalništvo in informatiko Univerze v Ljubljani raziskovalno delo v okviru projekta Obvladovanje procesa razvoja pri razvoju rešitev za elektronsko poslovanje, katerega cilj je izdelati pristop in računalniško podporo za načrtovanje in razvoj metodologij, ki so tako tehnično kot tudi socialno ustrezne posameznim organizacijskim sistemom oziroma njihovim projektom. V ta namen je bil zasnovan scenarij za načrtovanje in razvoj prilagodljive metodologije razvoja informacijskega sistema, ki obsega aktivnosti, za katere je bilo treba v okviru projekta CO ICT izboljšati metodološko podporo in izdelati programske module za podporo njihove avtomatizacije. Omenjeni programski moduli predstavljajo sistemsko podporo za delo z metodologijo in tvorijo obširnejšo aplikacijo – AMT (angl. Agile Methodology Toolset).

V prispevku najprej predstavimo scenarij za razvoj in načrtovanje prilagodljive metodologije, programske module aplikacije AMT, njihove funkcije in bistvene tehnične značilnosti celotne aplikacije (poglavje 2). Osrednji del prispevka je namenjen predstavitvi programskega modula MethAdapt, ki omogoča prilagoditi proces razvoja informacijskega sistema konkretnim potrebam projektov. V okviru tega predstavimo inovativen pristop, ki temelji na uporabi odločitvenih pravil za iskanje optimalne poti v grafu osnovnega procesa (poglavje 3). V sklepnem delu prispevka predstavimo še možno usmeritev za nadaljnje delo.

## 2 Aplikacija AMT

Namen izdelave programskega orodja je pomagati metodologu pri delu z metodologijami. Postopki izbire, prilagajanja in ugotavljanja sprejetosti metodologij so namreč zelo kompleksni in brez ustrezne računalniške podpore v praksi zelo težko izvedljivi. Z us-

trezno programsko podporo je tako mogoče zagotoviti hitrejše in učinkovitejše izvajanje aktivnosti, za katere je odgovoren metodolog, hkrati pa ga razbremenimo tudi zahtev glede dodatnih znanj s področij organizacijskih in socioloških ved, ki so potrebna za učinkovito izvedbo postopkov izbire in prilagajanja metodologij za potrebe organizacijskega sistema.

Slika 1 prikazuje scenarij za načrtovanje in razvoj prilagodljive metodologije razvoja informacijskega sistema v organizacijskem sistemu. V okviru scenarija so prikazane izbrane aktivnosti, za katere smo v okviru projekta CO ICT izboljšali metodološko podporo in izdelali programske module za podporo njihove avtomatizacije, ki tvorijo aplikacijo AMT.

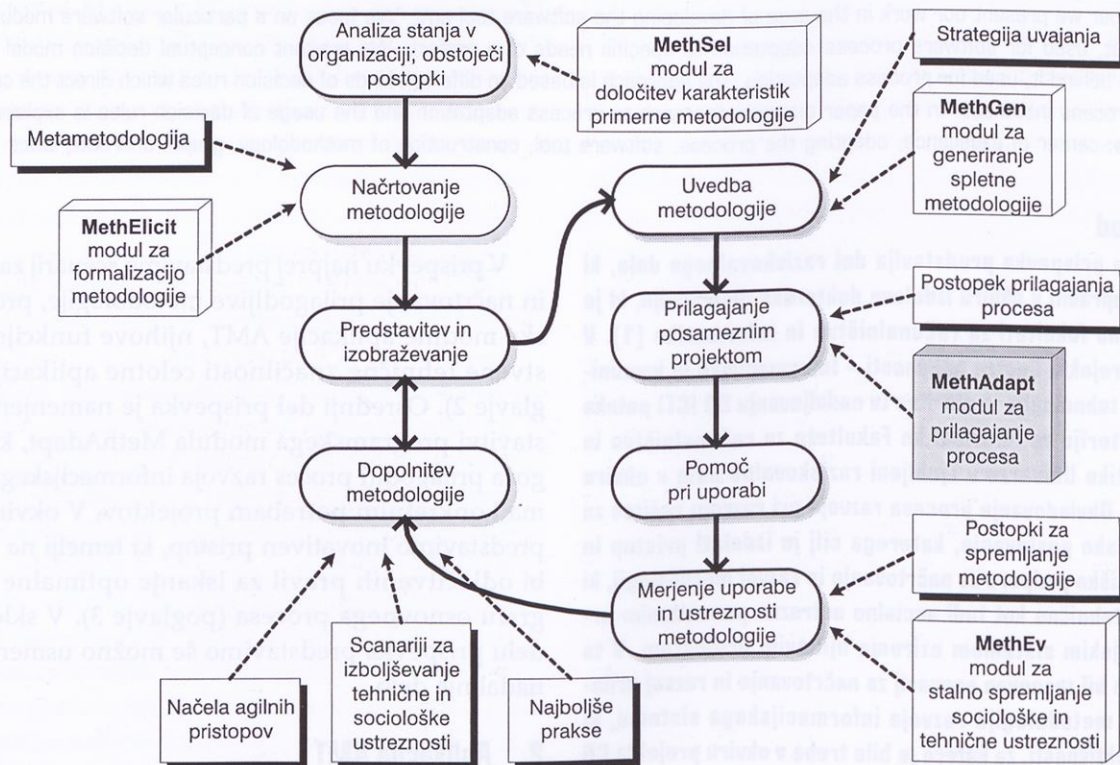
### 2.1 Programski moduli v AMT

Aplikacija AMT je programsko orodje, ki je namenjeno zajemu, prilagajanju in spremljanju tehnične in socialne ustreznosti metodologij razvoja informacijskega sistema. Sestavljena je iz naslednjih modulov:

- **MethElicit:** Modul za zajem in formalizacijo metodologije.
- **MethSel:** Modul za opredelitev značilnosti metodologije, ki bi bila primerna za uporabo z vidika celotnega organizacijskega sistema.
- **MethGen:** Modul za avtomatsko generiranje prikaza metodologije, primerne za uporabo na spletu.
- **MethAdapt:** Modul za podporo prilagajanja procesa v organizacijskem sistemu potrebam projektov, ki se izvajajo v njegovem okrilju.
- **MethEv:** Modul za stalno spremljanje sociološke in tehnične ustreznosti v organizacijski sistem vpeljane metodologije.

### 2.2 Tehnične lastnosti aplikacije AMT

AMT smo zasnovali kot spletno aplikacijo, ki teče na spletnem strežniku. Podatki, ki jih uporablja aplikacija, se hranijo na podatkovnem strežniku, do aplikacije pa se dostopa s katerim koli spletnim odjemalcem, ki



LEGENDA

- Aktivnost
- Sistemska podpora
- ▭ Metodološka podpora

Slika 1: Scenarij za načrtovanje in razvoj prilagodljive metodologije

podpira skriptni jezik Java in aktivne strežniške strani. Gre torej za uporabo klasične trinivojske arhitekture, za katero smo se odločili na podlagi potreb po zagotavljanju enostavne uporabe in enostavnosti dostopanja do same aplikacije (ne potrebujemo namenskega odjemalca). Aplikacijo AMT smo implementirali v Microsoftovi tehnologiji .NET, za podatkovni strežnik pa smo predvideli uporabo podatkovne zbirke SQL Server 2000.

### 3 Programski modul MethAdapt

Modul MethAdapt (na sliki 1 osenčen) predstavlja interaktivno orodje v sklopu aplikacije AMT, ki ga pri svojem delu uporabljajo metodologi. Delovanje modula [1] temelji na uporabi odločitvenega modela, katerega konceptualno zgradbo prikazuje slika 2.

Osrednji element odločitvenega modela predstavlja t. i. mehanizem prilagajanja, ki deluje na temelju uporabe odločitvenih pravil. Omenjeni element implementira idejo o predlaganem pristopu za prilagajanje procesa, zato ga bomo v nadaljevanju podrobneje predstavili.

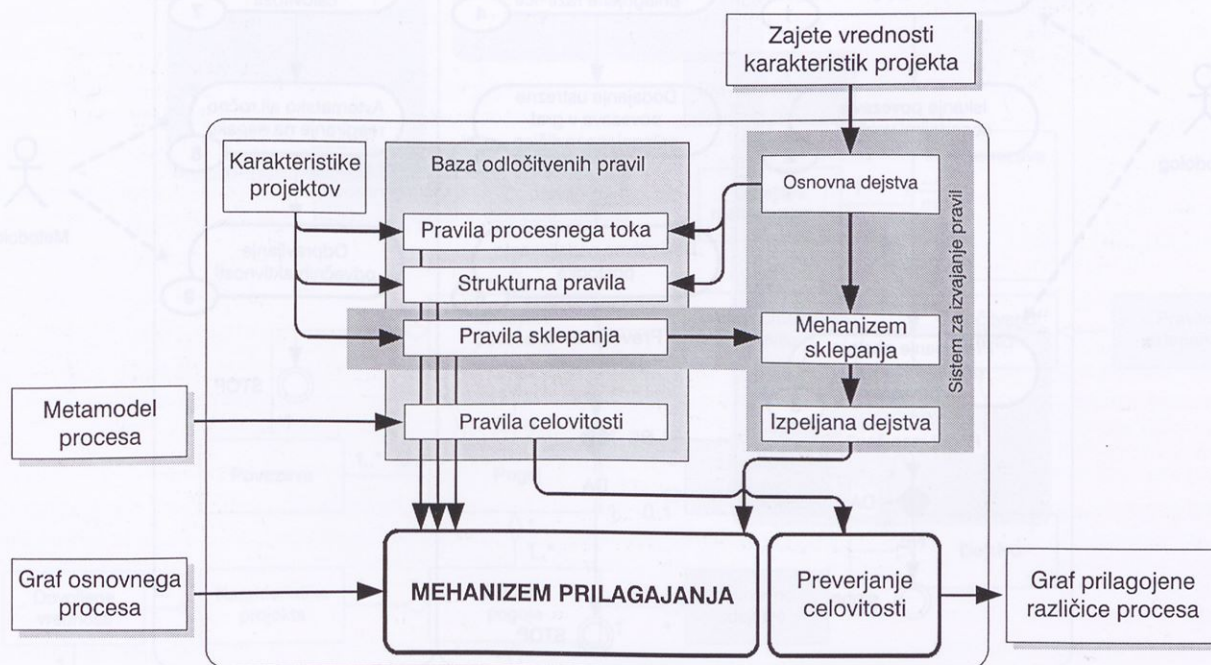
#### 3.1 Mehanizem prilagajanja

Ideja o prilagajanju procesa temelji na izsledkih raziskav s področja situacijskega konstruiranja metod-

ologij [2]. To področje opredeljuje več pristopov k prilagajanju metodologije in s tem razvojnega procesa, ki ga ta predpisuje. V okviru razvoja modula MethAdapt smo izhajali iz pristopa, ki temelji na predpostavki, da je prilagajanje splošne namenske metodologije možno izvesti z izbiro ustrezne, vnaprej določene poti skozi metodologijo. Omenjeni pristop smo razširili z odpravo te omejitve (omejeno število vnaprej določenih poti – prehodov skozi metodologijo). Pot skozi metodologijo se tako opredeli skozi postopek prilagajanja procesa in ta, skonstruirana pot, predstavlja prilagojeno različico procesa. Na ta način se lahko sam proces, ki ga predpisuje metodologija, natančneje prilagodi potrebam projekta, za katerega se izvede prilagajanje.

Mehanizem prilagajanja [1] smo opredelili kot rekurzivni postopek, ki v grafu splošnega procesa  $G$  poišče podgraf prilagojene različice  $G'$ , ob uporabi pravil procesnega toka, strukturnih pravil, pravil sklepanja ter izpeljanih dejstev.

V okviru mehanizma prilagajanja se nahaja tudi element za preverjanje celovitosti. Ta opredeljuje način za preverjanje celovitosti prilagojene različice procesa na temelju ugotavljanja spoštovanja omejitev, ki jih predpisuje metamodel uporabljene metodologije.



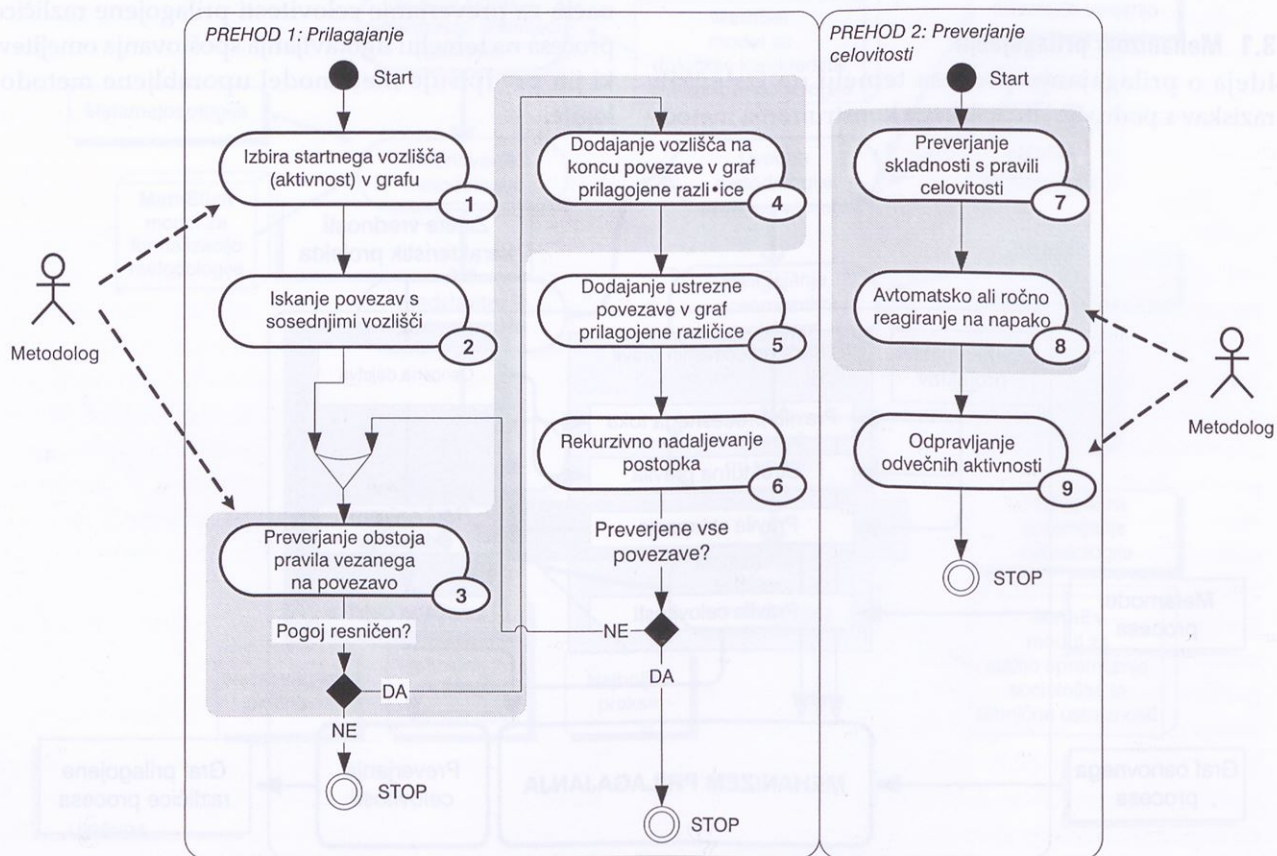
Slika 2: Konceptualna zgradba odločitvenega modela

Element za preverjanje celovitosti (v sklopu mehanizma prilagajanja) smo opredelili kot postopek, ki temelji na uporabi pravil celovitosti in tvori sestavni del postopka za gradnjo prilagojenih različic. Element odločitvenega modela – mehanizem prilagajanja – je opredeljen s postopkom, ki ga prikazuje diagram aktivnosti na sliki 3.

Postopek predvideva izgradnjo različice procesa v dveh prehodih skozi graf. S prvim je mišljen prehod skozi graf osnovnega procesa  $G$ , z drugim pa prehod skozi graf prilagojene različice  $G'$ . Vhod v postopek predstavlja graf osnovnega procesa  $G$ . V okviru prvega prehoda je treba v grafu najprej opredeliti startno vozlišče (tipa aktivnost), ki ga določi metodolog (aktivnost 1). V aktivnosti 2 se poišče povezave obravnavanega vozlišča s sosednjimi vozlišči. 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 obravnavani povezavi odločitveno pravilo pripisano

(strukturno pravilo ali pravilo procesnega toka) in je pogoj v njegovem pogojnem delu resničen, se vozlišče iz grafa  $G$ , ki se nahaja na koncu omenjene povezave in je hkrati opredeljeno tudi v posledičnem delu odločitvenega pravila, doda v graf prilagojene različice (aktivnost 4). V graf prilagojene različice  $G'$  se zatem doda tudi obravnavana povezava (aktivnost 5). V nadaljevanju postopka sledi rekurzivni klic, katerega parameter predstavlja vozlišče, ki je bilo dodano v graf prilagojenega procesa (aktivnost 6). Ko postopek preveri vse povezave, ki vodijo iz trenutnega vozlišča, se konča.

Izhod postopka predstavlja graf prilagojene različice  $G'$ . Vozlišča tega grafa predstavljajo elemente procesa (aktivnost, izdelek, orodje, tehnika itd.), ki tvorijo različico procesa, prilagojeno potrebam konkretnega projekta, ki pa še ni nujno celovita. Zaradi nepopolnosti zajetih vrednosti karakteristik konkretnega projekta se lahko zgodi, da postopek prilagajanja ne najde vseh vozlišč (elementov procesa), ki bi sicer



Slika 3: Postopek za prilagajanje procesa

morali biti vsebovani v prilagojeni različici, glede na omejitve, opredeljene v metamodelu metodologije. Problem smo rešili z opredelitvijo pravil celovitosti.

V okviru drugega prehoda skozi graf prilagojene različice  $G'$  postopek zagotovi celovitost prilagojene različice. Najprej se preveri skladnost prilagojene različice procesa s pravili celovitosti (aktivnost 7). V primeru neskladja z omejitvami iz metamodela se lahko z uporabo pravila celovitosti sproži akcija, ki jo pravilo predpisuje (aktivnost 8). Akcije v posledičnem delu pravil celovitosti lahko predlagajo različne načine za reševanje problema celovitosti. Vse aktivnosti v sklopu predstavljenega postopka, v katerih pride do izvajanja odločitvenih pravil, so na sliki 3 osenčene.

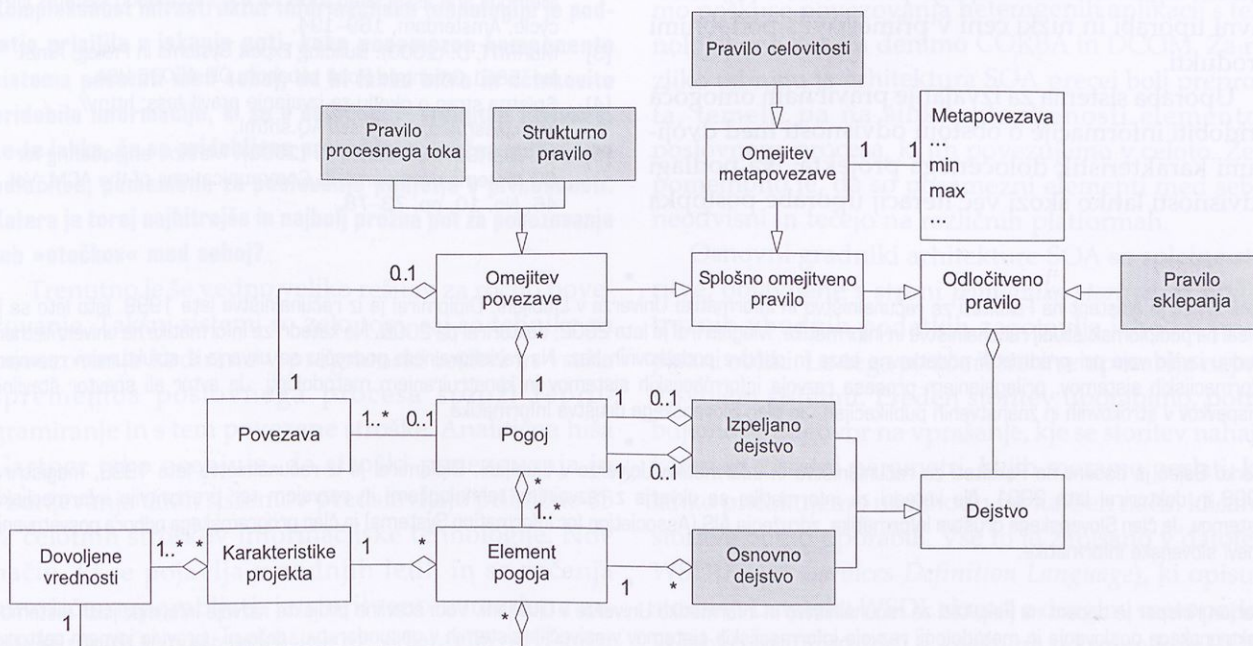
### 3.2 Odločitvena pravila

Odločitvena pravila [1], njihove komponente, povezave in druge elemente, na katere se navezujejo odločitvena pravila in smo jih uporabili v okviru predlaganega pristopa, prikazemo z uporabo metamodela pravil (slika 4).

Osrednji element metamodela – *odločitveno pravilo* – je predstavljen kot generalizacija *dejstev*, *splošnih omejitvenih pravil* in *pravil sklepanja*. Splošno omejitveno pravilo predstavlja generalizacijo pravil, ki so opredeljena na temelju *omejitev metapovezav* v meta-

modelu splošnega procesa in pravil, ki opredeljujejo *omejitve na povezavah* v grafu splošnega procesa. Pravila za opredelitev omejitev na povezavah v grafu predstavljajo generalizacijo *pravil procesnega toka* in *strukturnih pravil*. Iz omejitev posameznih metapovezav pa izvirajo opredelitve *pravil celovitosti*. Vsaki *metapovezavi* pripada določena omejitev metapovezave, ki je opredeljena na podlagi minimalne in maksimalne števности te povezave.

*Dejstvo* je predstavljeno kot generalizacija *osnovnega dejstva* in *izpeljanega dejstva*. Osnovno dejstvo je sestavljeno iz enega elementa pogoja, ki ga sestavlja natančno ena *karakteristika projekta*. Za vsako karakteristiko projekta je opredeljena vsaj ena dovoljena vrednost in vsaka dovoljena vrednost se lahko nahaja v več elementih pogoja. Element pogoja lahko pripada več *pogojem*. Isti pogoj se lahko uporabi v okviru več pravil za opredeljevanje omejitev povezave. Vsako pravilo za opredeljevanje omejitev povezav je pripisano natanko eni povezavi, medtem ko ni nujno, da je vsaki povezavi pripisano odločitveno pravilo. Velja tudi, da se isti pogoj nanaša na vsaj eno povezavo, medtem ko ni nujno, da se na določeno povezavo nanaša kakšen od pogojev. Izpeljana dejstva dobimo iz osnovnih dejstev. Pri tem ima bistveno vlogo mehanizem sklepanja (glej sliko 2), katerega delovanje temelji na pravilih sklepanja.



Slika 4: Metamodel odločitvenih pravil

Glede na povedano imajo uporabljena odločitvena pravila naslednjo zgradbo:

pravilo procesnega toka	IF <i>element procesa x</i> AND <i>pogoj</i> THEN <i>element procesa y</i>
strukturno pravilo	IF <i>element procesa x</i> AND <i>pogoj</i> THEN <i>element procesa y</i>
pravilo sklepanja	IF <i>pogoj</i> THEN <i>izpeljano dejstvo</i>
pravilo celovitosti	IF <i>metaelement procesa x</i> AND <i>metaelement procesa y</i> AND <i>pogoj</i> THEN <i>akcija</i>

Komponenta odločitvenega pravila *element procesa x* predstavlja začetno, *element procesa y* pa končno vozlišče povezave v grafu *G*, na katero se pravilo nanaša. Komponenta *pogoj* je sestavljena iz več elementov, ki so med seboj povezani z logičnimi operatorji konjunkcije, disjunkcije in negacije. Število elementov pogoja je odvisno od števila karakteristik projekta, ki vplivajo na vključitev elementa procesa v prilagojeno različico procesa, ki je predstavljen s končnim vozliščem povezave.

### 3.3 Sistem za izvajanje pravil

Pomembno komponento odločitvenega modela predstavlja tudi sistem za izvajanje pravil [3], za katerega realizacijo smo v okviru modula MethAdapt predvideli uporabo programskega okolja za izvajanje pravil Jess. Jess je sicer napisan v Javi, zaradi česar smo morali povezovanje z modulom MethAdapt realizirati prek spletne storitve, vendar temelji na zelo hitrem algoritmu za izvajanje pravil RETE [4], enostavni uporabi in nizki ceni v primerjavi s podobnimi produkti.

Uporaba sistema za izvajanje pravil nam omogoča pridobiti informacije o obstoju odvisnosti med dvojicami karakteristik določenega projekta. Na podlagi odvisnosti lahko skozi več iteracij uporabe postopka

pridemo do novega znanja, ki ga je mogoče uporabiti kot temelj za ažuriranje strukturnih pravil in pravil procesnega toka.

## 4 Sklep

Razvoj aplikacije AMT še ni končan. V naslednji fazi nas najprej čaka integracija posamičnih modulov v celoto in preizkus delovanja integrirane aplikacije. Med tem pa smo tudi že prišli do novih idej o nadaljnjem razvoju. Obstoječa zasnova AMT v okviru zajema elementov metodologije namreč ne omogoča gradnje metodologije iz standardiziranih elementov, za kar nam manjka repozitorij elementov [5]. Na področju konstruiranja metodologij se repozitorije zelo veliko omenja, manjkajo pa konkretne rešitve. Ključni problem pri tem predstavlja prav razvoj mehanizma za visokonivojsko klasifikacijo elementov, ki mora zagotoviti pravilen opis posameznega elementa metodologije, ki se nahaja v repozitoriju, tako da vemo, čemu je ta namenjen, ne da bi nam bilo treba pregledati njegovo specifikacijo. Razvoj repozitorija elementov zato vidimo kot možno smer za nadaljnji razvoj aplikacije AMT.

## 5 Viri in literatura

- [1] ZRNEC, Aljaž: Odločitveni model za prilagajanje procesa razvoja informacijskih sistemov individualnim potrebam projektov, doktorska disertacija, FRI 2006.
- [2] 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.
- [3] MERRITT, D. (2000). Building Expert Systems in Prolog. Amzi! inc. 5861 Greentree Road Lebanon, OH 45036 USA.
- [4] Spletna stran o okolju za izvajanje pravil Jess: <http://www.jessrules.com/jess/FAQ.shtml>.
- [5] HENDERSON-SELLERS, B. (2003). Method engineering for OO systems development. Communications of the ACM, Vol. 46, No. 10, pp. 73–78.

Aljaž Zrnc je asistent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Diplomiral je iz računalništva leta 1999. Isto leto se je vpisal na podiplomski študij računalništva in informatike. Magistriral je leta 2002, doktoriral pa 2006. Na katedri za informatiko na univerzitetnem študiju izvaja vaje pri predmetih podatkovne baze in osnove podatkovnih baz. Na raziskovalnem področju se ukvarja s strukturnim razvojem informacijskih sistemov, prilagajanjem procesa razvoja informacijskih sistemov in konstruiranjem metodologij. Je avtor ali soavtor številnih prispevkov v strokovnih in znanstvenih publikacijah. Je član Slovenskega društva Informatika.

Marko Bajec je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Diplomiral je iz računalništva leta 1996, magistriral 1998 in doktoriral leta 2001. Na katedri za informatiko se ukvarja z razvojnimi tehnologijami in razvojem ter prenovitvijo informacijskih sistemov. Je član Slovenskega društva Informatika, združenja AIS (Association for Information Systems) in član programskega odbora posvetovanja Dnevi slovenske informatike.

Marjan Krisper je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Vodi številne projekte razvoja informacijskih sistemov, elektronskega poslovanja in metodologij razvoja informacijskih sistemov v največjih sistemih v gospodarstvu, državni upravi in javnem sektorju. Je ustanovni član mednarodnega združenja za informacijske sisteme AIS (Association of Information Systems), član izvršnega odbora Slovenskega društva Informatika in član Slovenskega društva za umetno inteligenco.