

Analitski vzorci za poslovno-informacijske arhitekture

Ana Šaša, Marjan Krisper

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška 25, 1000 Ljubljana

ana.sasa@fri.uni-lj.si; marjan.krisper@fri.uni-lj.si

Izvleček

Poslovno-informacijske arhitekture so se v poslovnih sistemih izkazale kot sredstvo za učinkovitejše uresničevanje vizije in ciljev ter za zagotavljanje zveznosti in skladnosti posameznih delov poslovnega sistema. Pomembno področje poslovno-informacijskih arhitektur je doseganje skladnosti poslovnega in informacijskega sistema ter celostno obvladovanje informatike. Arhitekturna analiza je pri tem ena izmed ključnih aktivnosti. Predstavlja temelj za doseganje koristi, ki jih lahko poslovni sistem pridobi s poslovno-informacijsko arhitekturo. Analitske arhitekturne tehnike so podlaga načrtovanju in odločanju in se uporabljajo za ocenjevanje različic arhitekture, za bolj informirane odločitve in študijo vpliva sprememb v poslovno-informacijski arhitekturi. V prispevku predstavljamo vlogo arhitekturne analize na področju poslovno-informacijskih arhitektur in ogrodje poslovno-informacijskih arhitektur ArchiMate, na katerem temelji naše delo. Podajamo pregled analitskih tehnik ogrodja ArchiMate in predstavljamo predlog razširitve tehnik z množico osnovnih analitskih vzorcev, ki se lahko uporabijo za ugotavljanje bolj in manj primernih struktur. Pri tem je poudarjena ustreznost aplikativne podpore poslovnim procesom in podatkovnim objektom.

Ključne besede: poslovno-informacijska arhitektura, arhitekturna analiza, arhitekturni vzorec, poslovni proces, storitev.

Abstract

ANALYTICAL PATTERNS FOR ENTERPRISE ARCHITECTURES

In business systems, enterprise architectures have become an important means of facilitating fulfilment of business vision and goals, and of ensuring coherence and consistency of different parts of the business system. Important application domains of enterprise architectures are business-IT alignment and holistic IT governance. In this context, architectural analysis plays a central role and is the basis for achieving the potential enterprise architecture benefits. Architectural analysis techniques are the foundation for planning and decision making using enterprise architectures, and are used especially for evaluation of different versions of an architecture for more informed decisions, and cause-effect analysis in enterprise architectures. In this paper, we present the role of architecture analysis in the field of enterprise architectures and introduce the ArchiMate framework, which is the basis for our work. We give an overview of the ArchiMate analysis techniques and present an enhancement of architecture analysis efforts with a set of basic analytical patterns for a discovery of more or less suitable structures. The focus is on information system support for business processes and business objects.

Key words: enterprise architecture, architectural analysis, architectural pattern, business process, service.

1 UVOD

Poslovno-informacijske arhitekture predstavljajo bazo znanja poslovnega sistema, katere jedro zajema elemente notranjega in zunanjega okolja poslovnega sistema in povezave med njimi. Z naraščajočimi zahtevami po agilnosti poslovnih sistemov ter usklajenosti poslovnega sistema in informacijskih sistemov so poslovno-informacijske arhitekture (angl. *enterprise architectures*) postale zelo pomembno področje, ki mu veliko pozornosti posvečajo strokovnjaki in raziskovalci tako s področja informatike kot iz poslovne domene. Predstavljajo orodje za doseganje zveznosti in skladnosti posameznih delov poslovnega sistema, povezanosti strateških elementov s poslovnimi procesi, povezanosti poslanstva in poslovnih ciljev s cilji informatike ter za doseganje bolj in-

formiranih odločitev o nekaterih ključnih tematikah, kot so integracija informacijskih sistemov, povezovanje z zunanjimi poslovnimi in informacijskimi sistemi, optimizacija poslovnih procesov, obvladovanje poslovnih sprememb in sprememb informatike itn.

Osnova za koristi, ki jih lahko pridobi poslovni sistem s poslovno-informacijsko arhitekturo, je arhitekturna analiza. Različne tehnike arhitekturne analize so pomembne predvsem za podporo odločanju, za načrtovanje in za optimizacijo arhitekture. Vedno ko je potrebna sprememba v poslovno-informacijski arhitekturi, igra arhitekturna analiza pri tem osrednjo vlogo. Analitske arhitekturne tehnike se uporabljajo za ocenjevanje različic arhitekture, za bolj infor-

mirane odločitve pri sklepanju kompromisov, npr. med stroški, kvaliteto in učinkovitostjo, ter pri študiji vpliva sprememb v poslovno-informacijski arhitekturi [9]. Brez ustreznih analitskih tehnik poslovno-informacijska arhitektura služi predvsem za predstavitev in za komunikacijo, medtem ko so teže realizirana druga področja uporabe in s tem tudi koristi, ki bi jih lahko pomenila le-ta.

V prispevku podajamo predlog razširitve tehnik arhitekturne analize z množico osnovnih analitskih vzorcev, ki se lahko uporabijo za ugotavljanje bolj in manj ustreznih struktur v dani poslovno-informacijski arhitekturi. Pri tem je poudarek na ustreznosti aplikativne podpore poslovnim procesom in podatkovnim objektom ter na ponovni uporabi posameznih storitev in podatkovnih objektov. Obstoječe tehnike ne obsegajo tovrstnih analitskih vzorcev, kar analitikom otežuje pridobivanje pomembnih informacij o arhitekturnih rešitvah, ugotavljanje pomembnih lastnosti in odločanje. Vzorce lahko uporabimo kot podlago za opredelitev smernic, ki nam povedo, ali je obstoječa poslovno-informacijska arhitektura ustrezna, kaj bi lahko izboljšali, za ocenjevanje različnih ciljnih arhitektur in njihovo primerjavo. V predlogu analitskih vzorcev se osredinimo na arhitekturno analizo na podlagi ogrodja ArchiMate, saj gre za ogrodje, ki je usmerjeno k medsebojnim povezavam različnih arhitekturnih plasti. V prispevku se ne ukvarjamo z analitskimi tehnikami, ki se nanašajo samo na posamezno arhitekturno plast, saj se v ta namen lahko uporabljajo specializirane tehnike posameznih področij.

2 PREDSTAVITEV PODROČJA POSLOVNO-INFORMACIJSKIH ARHITEKTUR

Konceptualna osnova področja arhitektur je bila postavljena leta 2000 s sprejetjem standarda IEEE 1471-2000 (*IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*) [5]. Standard pomeni teoretično osnovo za definiranje, analizo in opis arhitekture sistemov, kot je informacijski sistem [11]. Standard podaja tole definicijo arhitekture sistema: *Arhitektura je ključni sestav sistema, ki vključuje njegove komponente, njihove medsebojne povezave in povezave z okoljem ter načela, ki vodijo njeno načrtovanje in razvoj.* (Standard IEEE 1471-2000, IEEE Computer Society, 2000 [5])

Na področju poslovno-informacijskih arhitektur še ne obstaja konsenz nad različnimi elementi in de-

finicijami, zato standard IEEE 1471-2000 še vedno predstavlja pomemben temelj tudi na tem področju. Prav tako obstaja več bolj ali manj različnih definicij izraza poslovno-informacijska arhitektura (PIA). Med njimi podajamo definicijo, ki jo je opredelila organizacija *The Open Group* in je med bolj razširjenimi: *Poslovno-informacijska arhitektura je formalen opis sistema ali podrobni plan sistema na nivoju komponent, ki usmerja njegovo implementacijo. Zajema strukturo komponent, njihovih medsebojnih povezav in načel ter smernic, ki vodijo njihovo načrtovanje in evolucijo skozi čas.* (*The Open Group*, 2009 [15])

Poslovno-informacijska arhitektura je eden od ključnih dejavnikov za zagotavljanje dolgoročne uspešnosti poslovnega sistema in je še posebno pomembna v kompleksnih poslovnih sistemih. Uporablja se predvsem za tri ključne namene, in sicer:

1) *kot osnova za predstavitev in komunikacijo:*

Poslovno-informacijska arhitektura daje celovit pogled na delovanje poslovnega sistema in njegovo sodelovanje navzven. Različni modeli, ki izhajajo iz poslovno-informacijske arhitekture, posameznim deležnikom predstavijo natančno točno tisti njen del, ki je zanje relevanten, in na način, ki ga umešča v celostni pogled na poslovni sistem. S tem so tudi podlaga za komunikacijo med različnimi deležniki;

2) *kot osnova za načrtovanje:*

Poslovno-informacijska arhitektura lahko zajema opis obstoječega stanja ali zelenega stanja. Pri tem lahko analiziramo različne variante in razhajanja med njimi – kaj je treba spremeniti, dodati, prilagoditi, da bi dosegli zeleno stanje. Pri tem igrajo pomembno vlogo tehnike arhitekturne analize, npr. analiza vpliva sprememb;

3) *za zagotavljanje skladnosti in zveznosti vseh delov poslovnega sistema:*

Poslovno-informacijska arhitektura omogoča zagotavljanje povezanosti poslanstva, vizije, poslovnih ciljev, poslovne strategije itn. s poslovnimi procesi in organizacijo. S tem so strategija in cilji posameznih delov poslovnega sistema usklajeni s strategijo in cilji celotnega poslovnega sistema, kar pomeni usmerjen fokus delovanja posameznih delov sistema pri uresničevanju strategije in poslanstva ter doseganju poslovnih ciljev in vizije.

Nekatere ključne koristi uporabe poslovno-informacijskih arhitektur lahko povzamemo v naslednjih točkah:

- poslovno-informacijska arhitektura daje celovit pogled na delovanje poslovnega sistema in njegovo sodelovanje navzven,
- zagotavlja zveznost in skladnost posameznih delov poslovnega sistema ter usmerjen fokus delovanja različnih delov poslovnega sistema k doseganju poslovnih ciljev ter vizije,
- zagotavlja povezanost poslanstva, vizije, poslovne strategije in poslovnih ciljev s poslovnimi procesi, z rezultati poslovnih procesov, z organizacijo poslovnega sistema,
- strategija in cilji informatike so usklajeni s poslovno strategijo in s poslovnimi cilji,
- je podlaga za optimizacijo poslovnih procesov,
- omogoča analizo vpliva sprememb (npr. kako se nov poslovni cilj odraža v izvajanju poslovnih procesov, v informacijski podpori poslovnih procesov, v organizacijski strukturi itn.),
- je podlaga za strateško planiranje tako poslovnega sistema kot njegovega informacijskega sistema,
- je podpora za odločanje,
- je sredstvo za komunikacijo in obvladovanje znanja v poslovnem sistemu,
- je podlaga za zagotavljanje interoperabilnosti,
- omogoča merjenje zmogljivosti in optimizacijo gradnikov arhitekture itn.

Za doseganje večine izmed potencialnih koristi poslovno-informacijskih arhitektur igra arhitekturna analiza osrednjo vlogo.

2.1 Pregled pristopov k poslovno-informacijskim arhitekturam

Najstarejše in med najbolj razširjenimi ogrodji poslovno-informacijskih arhitektur je Zachmanovo ogrodje, ki izhaja iz leta 1987 [16]. Zachmanovo ogrodje je shema, ki omogoča klasifikacijo artefaktov poslovno-informacijskih arhitektur in opisuje različne poglede deležnikov na poslovni sistem v skladu z njihovimi interesi. Jedro ogrodja je matrika dimenzije 6 x 6, ki identificira 36 pogledov na arhitekturo. Pri tem stolpci temeljijo na vprašalnicah: Kaj? Zakaj? Kdaj? Kdo? Kje? Zakaj? Vrstice temeljijo na konkretizaciji in predstavljajo različne zorne kote: identifikacija, opredelitev, predstavitev, specifikacija, konfiguracija in konkretizacija [17].

Drugi pomemben pristop, ki poslovno-informacijsko arhitekturo obravnava celostno, je TOGAF (*The Open Group Architecture Framework*). Razvija se

pod okriljem organizacije *The Open Group*¹ in zajema različne vidike, smernice, referenčne modele, meta-model in aktivnosti, ki so potrebni za zajem in vzdrževanje poslovno-informacijske arhitekture. Sestavljen je iz več delov, med katerimi jedro predstavlja arhitekturna metoda TOGAF ADM (*Architecture Development Method*). Pomembno razlikovanje med ogrodjem TOGAF in Zachmanovim ogrodjem je prav v metodi. Zachmanovo ogrodje namreč ne predpisuje, kako naj zajamemo ali vzdržujemo poslovno-informacijsko arhitekturo, temveč omogoča klasifikacijo arhitekturnih artefaktov po poljubnem pristopu.

Najnovejši pristop k poslovno-informacijskim arhitekturam je ArchiMate. Prav tako ga razvija organizacija *The Open Group* in zelo hitro pridobiva na pomenu. Podrobneje je predstavljen v razdelku 3 (Predstavitev ogrodja in analitskih tehnik ArchiMate).

Druga pomembnejša in zelo razširjena pristopa sta še FEAF (*Federal Enterprise Architecture Framework*) [1] in Gartnerjevo ogrodje za poslovno-informacijske arhitekture [6]. V posameznih okoljih se pri uvedbi poslovno-informacijskih arhitektur bodisi uporabi enega izmed splošno namenskih pristopov bodisi se ga prilagodi za specifično okolje ali se razvije lasten pristop.

2.2 Zorni koti in pogledi

Poslovno-informacijska arhitektura navadno opisuje veliko množico komponent in relacij med njimi. V poslovnem sistemu nastopajo različni akterji z različnimi vlogami. Ker se poslovno-informacijska arhitektura uporablja kot podlaga za predstavitve, komunikacijo, načrtovanje, analizo in odločanje, so posamezni modeli namenjeni različnim deležnikom z različnimi nalogami. Za vsakega izmed njih je relevanten le del poslovno-informacijske arhitekture. Modeli, ki bi vsebovali vse elemente in povezave med njimi, bi za posameznega deležnika vsebovali velik del informacij, ki so zanj nebitvene, postranskega pomena ali celo nepomembne. Poleg tega lahko na poslovno-informacijsko arhitekturo gledamo z različnih ravni podrobnosti. Za posameznega deležnika je ustrezna določena raven podrobnosti. To po-

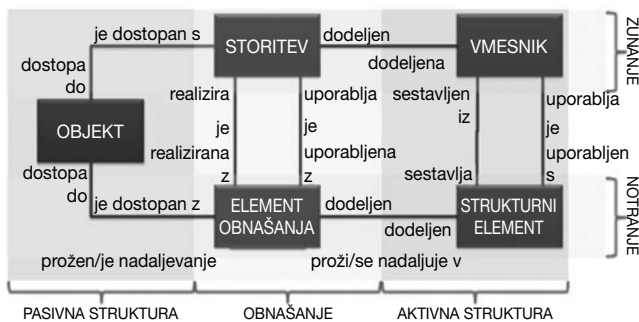
¹ *The Open Group* je organizacija, katere vizija je omogočanje na odprtih standardih in globalni interoperabilnosti temelječega dostopa do integriranih informacij znotraj in med poslovnimi sistemi. Je konzorcij, ki je neodvisen od ponudnikov rešitev. Njeni člani izhajajo iz vseh sektorjev skupnosti, povezanih z informacijsko tehnologijo – stranke, dobavitelji sistemov in rešitev, ponudniki orodij, integratorji, svetovalci, akademiki in raziskovalci.

meni, naj bi modeli za posameznega deležnika vsebovali natanko tiste elemente, ki jih zanimajo, na ustrezni ravni podrobnosti. V ta namen večina pristopov poslovno-informacijskih arhitektur opredeljuje zorne kote in poglede glede na posamezne deležnike. Zorni kot (ali vidik; angl. *viewpoint*) določa, kateri tipi elementov in na kateri ravni podrobnosti naj bodo vsebovani v modelu, namenjenemu danemu deležniku. Na podlagi zornega kota in danega opisa poslovno-informacijske arhitekture dobimo pogled na najo, ki je ustrezen za danega deležnika. Pogled je predstavitev sistema glede na zanimanja določenega deležnika.

3 PREDSTAVITEV OGRODJA IN ANALITSKIH TEHNIK ARCHIMATE

ArchiMate je najnovejši pristop k poslovno-informacijskim arhitekturam. Del pristopa sestavlja jezik ArchiMate, ki je bil kot tehnični standard sprejet v lanskem letu pri organizaciji *The Open Group* [13]. Ključni cilj ArchiMate je integracija arhitekturnih domen [9]. Zato ArchiMate določa *enoten jezik* za opisovanje strukture in delovanja poslovnih procesov, organizacijske strukture, informacijskih tokov, sistemov informacijske tehnologije in tehnične infrastrukture. Ključna motivacija je preseči razhajanja med različnimi arhitekturnimi domenami, ki navadno obstajajo v poslovnih sistemih, npr. med domenami poslovnih procesov, tehnične arhitekture in aplikativne arhitekture [14].

Elemente jezika ArchiMate lahko delimo v tri skupine, in sicer na aktivne strukturne elemente, elemente obnašanja in pasivne strukturne elemente (slika 1). Aktivni strukturni elementi so nosilci obnašanja, npr. poslovni akterji ali aplikativne komponente. Aktivnim strukturnim elementom so zato navadno dodeljeni elementi obnašanja, npr. določena funkcija je dodeljena poslovnemu akterju, ki je zadolžen za



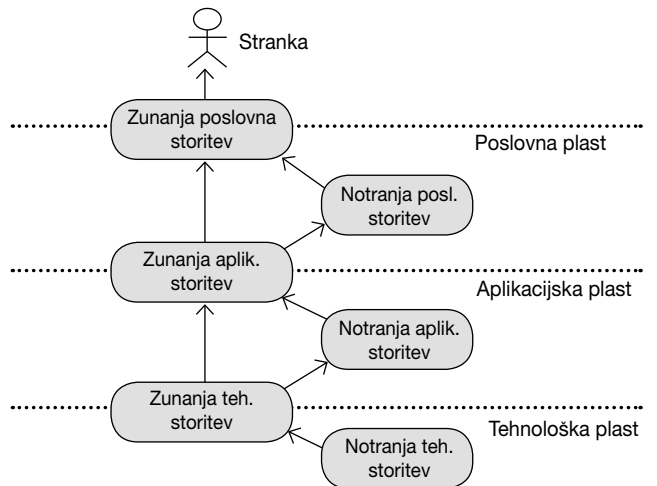
Slika 1: **Metamodel ključnih elementov ArchiMate**

njeno izvajanje. Pasivni strukturni elementi so objekti, na katerih se izvaja obnašanje ali so pod vplivom elementov obnašanja, npr. poslovni objekt, ki je ažuriran v poslovnem procesu.

Pri ArchiMateu je pomembno tudi razlikovanje med notranjim in zunanjim pogledom na sistem (slika 1); npr. pri vidiku obnašanja načelo zunanjega in notranjega pogleda odraža storitveno usmerjenost.

Koncept storitve v ogrodju ArchiMate igra osrednjo vlogo. Po ArchiMateu je storitev enota funkcionalnosti, ki jo dana entiteta (poslovni sistem, aplikativni sistem, organizacijska enota itn.) ponuja svojemu (zunanjemu ali notranjemu) okolju in ki eni ali več entiteta iz okolja predstavlja določeno vrednost [9] [12]. Storitve so lahko po naravi in po granularnosti zelo različne. Storitvena usmerjenost vodi v večplastni pogled na poslovno-informacijsko arhitekturo, pri čemer je storitev osrednja vez med različnimi plastmi. ArchiMate ločuje med tremi arhitekturnimi plastmi [12], in sicer med poslovno, aplikativno in tehnološko plastjo. Vsako plast deli v dve ravni, in sicer:

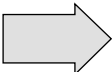
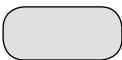

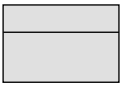








- na raven zunanjih storitev, ki obsega storitve, ki jih plast daje svojemu zunanjemu okolju in se uporabljajo na višjih arhitekturnih plasteh, in
- na implementacijsko raven, ki vsebuje komponente in relacije med njimi ter storitve, ki se uporabljajo znotraj posamezne plasti (notranje storitve).



Slika 2: **Večplastna arhitektura in koncept storitve [8]**

Implementacijska raven realizira storitveno raven. Slika 2 ponazarja posamezne plasti in njihovo povezovanje s konceptom storitve. Tabela 1 podaja legendo simbolov jezika, ki bodo uporabljeni v nadaljevanju prispevka.

Tabela 1: **Legenda uporabljenih simbolov jezika ArchiMate**

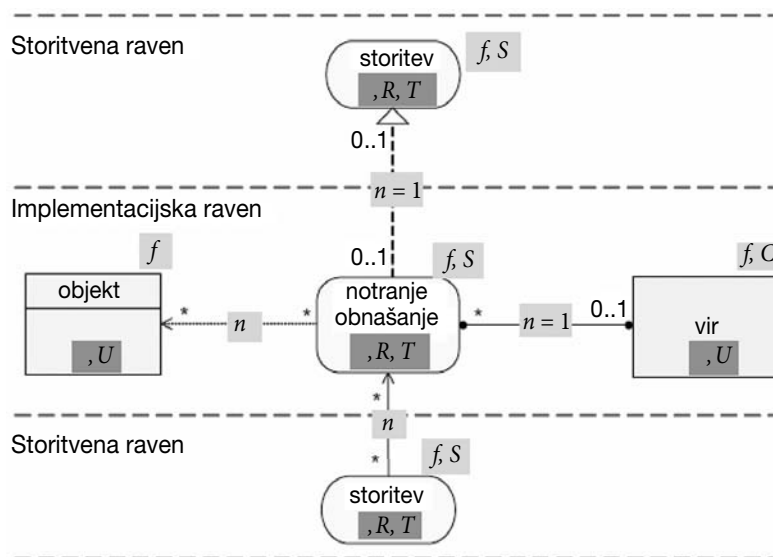
Simbol	Kratek opis
	Proces (Process) Zaporedje podprocesov oz. funkcij, ki vodijo do določenih rezultatov – izdelkov ali storitev.
	Storitev (Service) Navzven vidna enota funkcionalnosti, ki ima svoj pomen. Lahko gre za organizacijsko storitev (storitev, ki jo organizacija nudi svojemu okolju) ali aplikativno storitev (storitev, ki jo določen aplikativni sistem ponuja poslovnemu sistemu).
	Aplikativna komponenta (Application component) Modularni, zamenljivi del sistema, ki ponuja funkcionalnost prek vmesnikov. Z aplikativno komponento lahko predstavimo aplikativni sistem ali posamezne module aplikativnega sistema.
	Objekt (Object) Ločimo med dvema tipoma objektov: poslovni objekt predstavlja koncept, ki ima s poslovnega vidika določen pomen, podatkovni objekt pa predstavlja podatek ali množico podatkov, primerno za avtomatsko procesiranje – lahko je realizacija poslovnega objekta.
	Predstavitev (Representation) Predstavitev je zaznavna realizacija poslovnega objekta, npr. dokument v papirni obliki.
	Proženje (Triggering) Relacija med dvema entitetama obnašanja, npr. procesoma, ki pomeni, da konec prve entitete proži začetek druge.
	Realizacija (Realization) Relacija, ki prikazuje, kako so logične entitete, npr. storitve, realizirane z bolj oprijemljivimi entitetami, npr. z aplikativno komponento.
	Uporaba (Used-by) Relacija, ki prikazuje medsebojno uporabo gradnikov, npr. proces uporablja storitev.
	Dodelitev (Assignment) Relacija, ki se lahko uporablja za dodelitev določenega elementa obnašanja aplikativni komponenti ali funkciji. Če je npr. poslovni proces dodeljen aplikativni komponenti, to pomeni, da aplikativna komponenta avtomatizira proces.
	Dostop (Access) Relacija, ki prikazuje, kako proces, funkcija, storitev ali dogodek dostopajo do poslovnega ali podatkovnega objekta.
	Agregacija (Aggregation) Relacija, ki kaže, da objekt združuje skupino drugih objektov.
	Kompozicija (Composition) Relacija, ki kaže, da je objekt sestavljen iz drugih objektov.

3.1 Obstoječe tehnike arhitekturne analize pristopa ArchiMate

3.1.1 Kvantitativna arhitekturna analiza

Kvantitativna arhitekturna analiza se ukvarja s pove-zovanjem med različnimi elementi in plastmi poslovno-informacijske arhitekture s kvantitativnega vidika. Uporablja se za optimizacijo s kvantificiranjem učinka različic načrtovalskih odločitev, za pridobitev meril, ki podpirajo analizo vpliva sprememb, in za planiranje kapacitet, npr. koliko ljudi mora sodelovati, da se proces izvede v roku. Cilj kvantitativne analize po ArchiMatu je določiti naslednja performančna merila:

- delovna obremenitev (*workload*) ali λ_a za vsak element poslovno-informacijske arhitekture (vozlišče). Če nobeden izmed virov ni preobremenjen, je prepustnost (*throughput*) vsakega vozlišča enaka njegovi delovni obremenitvi;
 - čas procesiranja T_a in odzivni čas R_r za vsak element obnašanja;
 - uporaba U_r posameznega vira r .
- Izračun meril poteka na podlagi kvantitativnih vhodnih podatkov modela, in sicer (slika 3):
- za vsako relacijo tipa e uporaba (*Used-by*) in dostop (*Access*) je podana utež n_e , ki predstavlja povprečje uporabe in dostopov;
 - za vsak element obnašanja a je podan storitveni čas S_a , ki predstavlja notranji čas, ki ga sistem porabi za realizacijo storitve (tj. čas procesiranja brez časa, ki ga sistem porabi za čakanje drugih uporabljenih storitev). Pri tem se predvideva, da storitev podeduje storitvene časovne vrednosti elementa, ki jo realizira;
 - za vsak vir r je podana kapaciteta C_r ;
 - za vsako vozlišče a je podana frekvenca prihajanj f_a . Navadno so frekvence prihajanj specificirane na vrhni plasti modela, čeprav so lahko podane za poljubno vozlišče modela.
- Izračun meril poteka v treh korakih:
1. normalizacija vhodnega modela z uporabo transformacij modelov z namenom izdelave modela, ki je skladen s strukturo, predstavljeno na spodnji sliki (slika 3);
 2. izračun delovnih obremenitev λ od zgoraj navzdol (*top-down*);
 3. izračun performančnih meril T , U in R na podlagi pristopa od spodaj navzgor (*bottom-up*).

Slika 3: **Osnovni koncepti kvantitativne arhitekturne analize po pristopu ArchiMate**

3.1.2 Kvalitativna arhitekturna analiza

ArchiMate loči med dvema tipoma kvalitativne arhitekturne analize (ali funkcionalne arhitekturne analize), in sicer med statičnim ali strukturnim tipom ter dinamičnim tipom ali tipom obnašanja.

Pri statični analizi arhitektur se uporabljajo formalizmi opisne logike. Opisna logika in na njej temelječi jeziki za predstavitev znanja so prilagojeni za zajem znanja o konceptih in hierarhijah konceptov. Na področju poslovno-informacijskih arhitektur je najbolj pomembno področje uporabe opisne logike pri določanju vpliva spremembe na arhitekturo.

Tehnike dinamične kvalitativne analize arhitektur temeljijo na formalnih pristopih, kot so npr. procesna algebra in omrežja podatkovnih tokov. Pri tem se lahko identificirajo suboptimalni deli arhitekture, ki se nanašajo na logične napake v arhitekturi, npr. dve sočasno aktivni vlogi, katerih aktivnosti so nekompatibilne (npr. prepisovanje, brisanje/uničevanje narejenega). Dinamična kvalitativna analiza pripomore k izboljševanju konsistentnosti in se osredini na logiko modelov.

4 ANALITSKI VZORCI POSLOVNO-INFORMACIJSKE ARHITEKTURE

V razdelku podajamo predstavitev osnovnih vzorcev kvalitativne analize poslovno-informacijske arhitekture, ki smo jih razvili z namenom formalizacije ocenjevanja poslovno-informacijskih arhitektur. Na-

našajo se na statične in dinamične vidike kvalitativne analize. Predstavljajo podlago za vizualizacijo različnih arhitekturnih struktur in za poizvedbe po neoptimalnih in drugih arhitekturnih strukturah.

Poslovno-informacijske arhitekture podjetij in drugih poslovnih sistemov so navadno kompleksne in obsegajo veliko število elementov in relacij med njimi. Osrednji del informacijskih sistemov za poslovno-informacijske arhitekture je navadno repozitorij, ki vsebuje informacije o elementih, o njihovih atributih in medsebojnih povezavah med elementi. Pri uporabi tovrstnih sistemov smo ugotovili, da je eden izmed večjih problemov pri analiziranju poslovno-informacijskih arhitektur v tem, da na voljo ni pristopa, ki bi opredeljeval, katere značilnosti poslovno-informacijskih arhitektur bi bilo treba obravnavati, kakšne informacije nosijo ter kako te poiskati iz kompleksnih repozitorijev poslovno-informacijskih arhitektur. Kljub temu da poslovno-informacijske arhitekture zajemajo koristne informacije, se zaradi pomanjkanja tovrstnih mehanizmov uporabniki srečujejo s številnimi problemi, kot so težave pri iskanju relevantnih informacij, ki bi koristile pri ocenjevanju obstoječe arhitekture, nepopolne informacije pri planiranju ciljne arhitekturne rešitve in težave pri primerjavi različnih potencialnih rešitev. Za zmanjšanje tovrstnih težav in omogočanje bolj informiranih odločitev ter argumentov pri analizi obstoječe in načrtovanju ciljne poslovno-informacijske arhitektu-

re predlagamo uporabo analitskih vzorcev. V prispevku se osredinjamo na analizo podpore poslovnim procesom in poslovnim objektom s pomočjo poslovno-informacijskih arhitektur. Pristop smo razvili na podlagi izkušenj, pogostih praks in težav, v katerih smo se srečevali na projektih v petih večjih slovenskih poslovnih sistemih. Cilj prispevka je predstaviti, kako lahko opredelimo karakteristike poslovno-informacijskih arhitektur, ki naslavljajo vidika podpore poslovnim procesom in poslovnim objektom, in omogočiti učinkovit način za njihovo prepoznavanje v kompleksnih repozitorijih poslovno-informacijskih arhitektur. Za opredelitev teh karakteristik uporabljamo koncept vzorcev.

Vzorec je *abstrakcija od konkretnega, ki se pojavlja v določenih nepoljubnih kontekstih* [10]. V danem prispevku izraz analitski vzorec pomeni *množico elementov poslovno-informacijske arhitekture, ki odražajo arhitekturne strukture z določenim pomenom za analitika*. Pri tem ne gre za klasične načrtovalske ali analitske vzorce, kot se pogosto uporabljajo v informatiki in računalništvu, saj niso namenjeni reševanju problemov z načrtovanjem aplikativnih sistemov ali s poslovnim modeliranjem [3][4]. Naš pristop temelji na odkrivanju vzorcev poslovno-informacijske arhitekture. Cilj je prepoznati vzorce, ki se pojavljajo v poslovno-informacijski arhitekturi: ugotovljeni vzorci za dani element nosijo informacije, ki jih analitik lahko uporabi kot vhod za arhitekturno analizo. Informacije lahko uporabi za različne namene, npr. za ugotavljanje, ali trenutna rešitev podpira poslovne cilje in zahteve, za določanje prednosti in slabosti obstoječe poslovno-informacijske arhitekture in za ugotavljanje, ali je primerna z vidika podpore poslovnim procesom oz. jo je treba izboljšati. Posamezen vzorec v prispevku predstavljamo kot množico elementov, ki jo formaliziramo z nujnimi in zadostnimi pogoji pripadnosti množici (pogoji pripadnosti vzorca). Če določen element poslovno-informacijske arhitekture izpolnjuje pogoje pripadnosti vzorca, potem pravimo, da ustreza vzorcu ali da izkazuje vzorec.

Zaradi kompleksnih množic elementov in njihovih medsebojnih povezav v poslovno-informacijski arhitekturi je bil eden izmed ciljev opredeliti vzorce na način, ki omogoča njihovo implementacijo v informacijskih sistemih za poslovno-informacijsko arhitekturo. Vzorci so formalno opredeljeni s pogoji pripadnosti, kar omogoča ne le uporabo tehnik za njihovo zaznavanje, temveč tudi njihovo implementacijo v

orodjih za zajem in vzdrževanje poslovno-informacijskih arhitektur, ki podpirajo jezik ArchiMate in omogočajo uporabo skriptnega ali poizvedbenega jezika. Zadnje velja za večino obstoječih orodij, ki podpirajo ArchiMate, npr. za BiZZdesign Architect (BiZZdesign), ARIS ArchiMate Modeler (IDS Scheer AG) in IBM Rational System Architect (IBM). Podpora vzorcem v orodju za zajem in vzdrževanje poslovno-informacijske arhitekture lahko služi za različne primere uporabe, npr. v obliki avtomatskih opozoril, če so zaznane neoptimalne strukture, ali v obliki podpore odločanju pri primerjavi različnih potencialnih ciljnih rešitev.

V nadaljevanju so najprej podani simboli, ki se bodo uporabljali za opredelitev vzorcev, nato pa so opredeljeni analitski vzorci. Posamezen analitski vzorec je podan z opisom, pogoji pripadnosti in primerom v jeziku ArchiMate.

4.1 Opredelitev uporabljenih simbolov

Tabela 2: **Opredelitev uporabljenih simbolov**

BP	Množica vseh poslovnih procesov
BF	Množica vseh poslovnih funkcij
BI	Množica vseh poslovnih interakcij
BE	Množica vseh poslovnih dogodkov
BO	Množica vseh poslovnih objektov
DO	Množica vseh podatkovnih objektov
R	Množica vseh predstavitev
OS	Množica vseh organizacijskih storitev
AS	Množica vseh aplikativnih storitev
TS	Množica vseh tehnoloških storitev
AK	Množica vseh aplikativnih komponent
$AnBP$	Množica analiziranih poslovnih procesov: $AnBP: AnBP \subseteq BP$
$AnBO$	Množica analiziranih poslovnih objektov: $AnBO: AnBO \subseteq BO$
BBE	Množica vseh elementov obnašanja: $BBE = BP \cup BF \cup BI \cup BE$
X^+	Tranzitivno zaprtje relacije X

Opomba: Posamezne množice se nanašajo na elemente v dani poslovno-informacijski arhitekturi.

Posamezne relacije, ki jih navajamo, vključujejo tudi tranzitivno izpeljane relacije.

4.2 Vzorci aplikativne podpore poslovnih procesov

V arhitekturnem opisu v jeziku ArchiMate je podpora poslovnih procesov zajeta z relacijama uporaba in dodeljevanje. Z vidika avtomatiziranosti in podpore poslovnih procesov ločimo med naslednjimi vzorci aplikativne podpore poslovnih procesov:

- vzorec avtomatiziranega poslovnega procesa: poslovni proces je dodeljen aplikativni komponenti, ki ga popolnoma avtomatizira;
- vzorec delno avtomatiziranega poslovnega procesa: poslovni proces uporablja informacijski sistem, ki avtomatizira oz. podpira določene aktivnosti v procesu. Preostale aktivnosti izvajajo zaposleni v poslovnem sistemu;
- vzorec aplikativno nepodprtega (neavtomatiziranega) poslovnega procesa: poslovni proces nima podpore v informacijskem sistemu;
- vzorec neaplikativno podprtega poslovnega procesa: poslovni proces izvajajo poslovni akterji brez aplikativne podpore;
- vzorec (strogo) nepodprtega poslovnega procesa: poslovni proces nima ne aplikativne podpore niti ga ne izvaja poslovni akter. Gre lahko bodisi za procese, ki jih planiramo in aplikativna podpora ter odgovorne vloge še niso določene, ali za procese, ki jih opuščamo.

Glede na raznolikost aplikativnih sistemov, ki lahko nastopajo v poslovnem procesu, ločimo dva vzorca aplikativne podpore poslovnega procesa:

- vzorec heterogene aplikativne podpore poslovnega procesa: poslovni proces podpira več različnih aplikativnih sistemov;
- vzorec homogene aplikativne podpore poslovnega procesa: poslovni proces podpira en aplikativni sistem.

Ločimo tudi med različnimi vzorci aplikativne podpore posameznih procesnih aktivnosti. Ti v pri-spevku niso obravnavani.

Določen poslovni proces lahko izkazuje več vzorcev. S pomočjo vzorcev aplikativne podpore poslovnih procesov lahko ugotavljamo stopnjo aplikativne podprtosti poslovnih procesov in njihovih podprocesov. Glede na naravo procesnih aktivnosti lahko analitik informacije uporabi za presojo, ali je smiselno dvigniti stopnjo aplikativne podpore ali ne, ter kot podlago za nadaljnje analitske aktivnosti, kot npr. za analizo vpliva spremembe ob dvigu stopnje podpore.

V nadaljevanju podrobneje predstavljamo nekatere izmed zgoraj navedenih vzorcev. Primeri, ki so podani za posamezne vzorce, se nanašajo na zorni kot aplikativne podpore analiziranih poslovnih procesov: vsebujejo analizirane poslovne procese, aplikativne storitve, ki jih uporabljajo ti procesi, aplikativne komponente, ki realizirajo storitve, ter vloge in aplikativne komponente, ki so jim dodeljeni procesi. Ne prikazujejo npr. drugih aplikativnih storitev, ki se ne uporabljajo v analiziranih procesih. Prav tako, razen relacij uporabe aplikativne storitve, dodelitve ter realizacije aplikativnih storitev ne prikazujejo drugih relacij med posameznimi elementi, četudi morda obstajajo v dani poslovno-informacijski arhitekturi.

4.2.1 Vzorec avtomatiziranih poslovnih procesov

Vzorec naslavlja poslovne procese, ki so v celoti avtomatizirani in ne vsebujejo aktivnosti, ki jih izvajajo ljudje. Če je poslovni proces avtomatiziran, potem je dodeljen vsaj eni aplikativni komponenti.

Naj funkcija $SP(bp)$ dani poslovni proces $bp \in BP$ preslika v množico njegovih podelementov obnašanja: podprocesov, poslovnih funkcij, poslovnih dogodkov in poslovnih interakcij, ki so del poslovnega procesa bp :

$$SP(bp) = \{sp \mid | sp \in BBE : (bp, sp) \in [composition]^{\uparrow} + \vee (bp, sp) \in Aggregation'\}$$

(FD1)

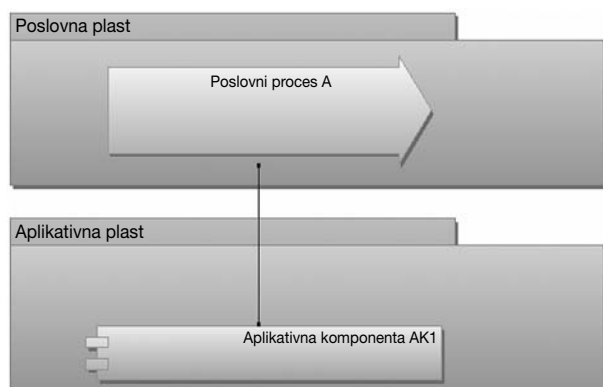
Avtomatizirane poslovne procese lahko opredelimo z naslednjo množico:

$$[ABP] \downarrow AnBP = \{a \mid | a \in AnBP \wedge ((\exists b \in AK : (b,a) \in Assignment) \vee (\forall sp \in SP(a) : \exists ac \in AK \wedge (ac, sp) \in Assignment))\}$$

(FD2)

Slika 4 prikazuje primer pogleda aplikativne podpore analiziranega poslovnega procesa A, ki je po-

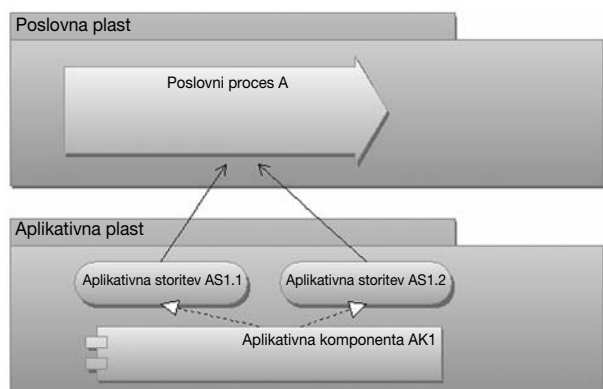
polnoma avtomatiziran z aplikativno komponento AK1.



Slika 4: **Aplikativno podprt poslovni proces – popolna avtomatizacija**

$$[SABP] \downarrow AnBP = \{a \mid a \in AnBP [ABP] \downarrow AnBP \wedge ((\exists b \in AS : (b,a) \in UsedBy) \vee (\exists sp1 \in SP(a), \exists ac \in AK : (ac, sp1) \in Assignment) \vee (\exists sp2 \in SP(a), \exists as \in AS : (as, sp2) \in UsedBy))\} \tag{FD3}$$

Slika 5 prikazuje primer pogleda aplikativne podpore poslovnega procesa A, ki je delno avtomatiziran z uporabo dveh aplikativnih storitev.



Slika 5: **Aplikativno podprt poslovni proces – delna avtomatizacija**

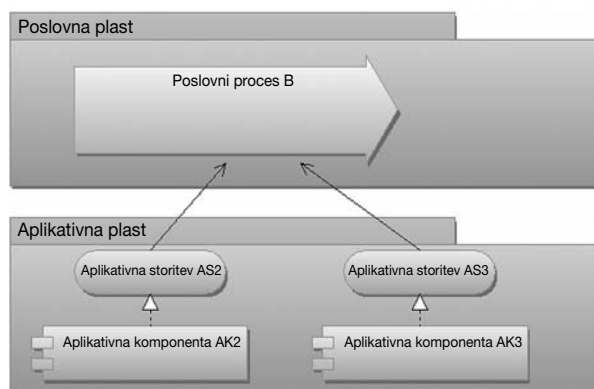
4.2.3 Vzorec heterogene aplikativne podpore poslovnega procesa

Poslovni proces s heterogeno aplikativno podporo je podprt z dvema različnima aplikativnima sistemoma ali več. Heterogena aplikativna podpora se nanaša na tiste delno avtomatizirane poslovne procese, pri kate-

4.2.2 Vzorec delno avtomatiziranih poslovnih procesov

Poslovni proces je delno avtomatiziran, če ni polno avtomatiziran (vzorec avtomatiziranih poslovnih procesov) in drži vsaj ena izmed naslednjih trditev: 1) poslovni proces uporablja vsaj eno aplikativno storitev, 2) vsaj eden izmed njegovih podelementov obnašanja je dodeljen aplikativni komponenti ali uporablja aplikativno storitev.

rih uporabniki pri izvajanju procesa prehajajo med več različnimi aplikativnimi sistemi. Npr. pri izvedbi prvega dela procesa delajo s prvim aplikativnim sistemom, pri izvedbi drugega dela procesa z drugim aplikativnim sistemom in pri izvedbi tretjega dela procesa s tretjim. To med drugim pomeni tudi, da ni podprt tok poslovnega procesa. Gre za neoptimalno strukturo, ki lahko potencialno povzroča ozka grla in odvečno delo. Heterogena aplikativna podpora se nanaša tudi na avtomatizirane poslovne procese, ki so dodeljeni več kot eni aplikativni komponenti.



Slika 6: **Heterogena aplikativna podpora poslovnega procesa**

Za nadaljnje opredelitve vzorcev opredelimo tri funkcije:

- $ACU(bp)$: funkcija, ki dani poslovni proces $bp \in BP$

preslika v množico aplikativnih komponent, ki se uporabljajo v procesu ali v njegovih podelementih obnašanja:

$$ACU(bp) = \{ac \mid ac \in AK \wedge (((ac, bp) \in UsedBy) \vee (\exists sp \in SP(bp) : (ac, sp) \in UsedBy))\} \quad (FD4)$$

- $ACA(bp)$: funkcija, ki poslovni proces $bp \in BP$ preslika v množico aplikativnih komponent, katerim je dodeljen proces:

$$ACA(bp) = \{ac \mid ac \in AK \wedge ((ac, bp) \in Assignment)\} \quad (FD5)$$

- $ACSPA(bp)$: funkcija, ki poslovni proces $bp \in BP$ preslika v množico aplikativnih komponent, ki so dodeljene podelementom obnašanja procesa bp :

$$ACSPA(bp) = \{ac \mid ac \in AK \wedge (\exists sp \in SP(bp) : (ac, sp) \in Assignment)\} \quad (FD6)$$

Poslovne procese, ki so heterogeno aplikativno podprti, lahko opredelimo z naslednjo množico:

$$[HeASBP] \downarrow AnBP = \{a \mid a \in AnBP \wedge |ACU(a) \cup ACA(a) \cup ACSPA(a)| \geq 2\} \quad (FD7)$$

Primer pogleda aplikativne podpore poslovnega procesa, ki kaže na heterogeno aplikativno podporo, prikazuje slika 6.

4.2.4 Vzorec homogene aplikativne podpore poslovnega procesa

Poslovni proces s homogeno aplikativno podporo je podprt z enim aplikativnim sistemom. Pri tem lahko aplikativni sistem, ki nudi podporo poslov-

nemu procesu, uporablja tudi druge aplikativne sisteme, vendar je to z vidika uporabnika transparentno. Avtomatizirani poslovni proces je vedno homogeno aplikativno podprt. Delno avtomatizirani poslovni proces je homogeno aplikativno podprt, če uporabniki pri izvajanju procesa uporabljajo en sam aplikativni sistem. Poslovne procese, ki so aplikativno homogeno podprti, opredelimo z množico:

$$[HoASBP] \downarrow AnBP = \{a \mid a \in AnBP \wedge (|ACU(a) \cup ACA(a) \cup ACSPA(a)| = 1)\} \quad (FD8)$$

Primer pogleda aplikativne podpore poslovnega procesa, ki kaže na homogeno aplikativno podporo, prikazujeta sliki 4 in 5.

4.3 Vzorci podpore poslovnih objektov

Z naraščajočimi količinami podatkov je pomembno, da so poslovni objekti podprti v informacijskem sistemu. V arhitekturnem opisu v jeziku ArchiMate je pod-

prtost poslovnih objektov zajeta z relacijo realizacija (*Realization*). Zato je osnovni vidik za analizo nepodprtosti poslovnih objektov vidik realizacije poslovnih objektov. Poslovni objekt je lahko realiziran bodisi s predstavitvijo ali s podatkovnim objektom. Ločimo med petimi vzorci podpore poslovnih objektov:

- vzorec (strogo) nepodprtih poslovnih objektov: poslovni objekt ni realiziran niti s predstavitvijo niti s podatkovnim objektom;
- vzorec neaplikativno podprtih poslovnih objektov: poslovni objekt ni podprt v informacijskem sistemu, vendar je realiziran s predstavitvijo, npr. z dokumentom v papirni obliki;
- vzorec delno neaplikativno podprtih poslovnih objektov: poslovni objekt je delno realiziran s predstavitvijo;
- vzorec aplikativno nepodprtih poslovnih objektov: poslovni objekt ni podprt z informacijskim sistemom;
- vzorec delno aplikativno podprtih poslovnih objektov: v informacijskem sistemu je poslovni objekt zajet (implementiran) le delno;
- vzorec aplikativno podprtih poslovnih objektov: poslovni objekt je v celoti implementiran v informacijskem sistemu.

Med podprtimi poslovnimi objekti lahko glede na števnost in vrsto realizacije (aplikativna, fizična) ločimo tudi med naslednjimi vzorci:

- vzorec večkratne realizacije poslovnega objekta,
- vzorec večkratne aplikativne realizacije poslovnega objekta,
- vzorec večkratne neaplikativne realizacije poslovnega objekta,
- vzorec enkratne realizacije poslovnega objekta,
- vzorec enkratne aplikativne realizacije poslovnega objekta,
- vzorec enkratne neaplikativne realizacije poslovnega objekta.

V nadaljevanju podrobneje predstavljamo nekatere izmed zgoraj navedenih vzorcev. Primeri, ki so podani za posamezne vzorce, se nanašajo na zorni kot aplikativne podpore in predstavitev analiziranih poslovnih objektov: vsebujejo analizirane poslovne objekte, poslovne objekte, ki jih sestavljajo ali agregirajo, ter podatkovne objekte in predstavitve, ki realizirajo dane poslovne objekte.

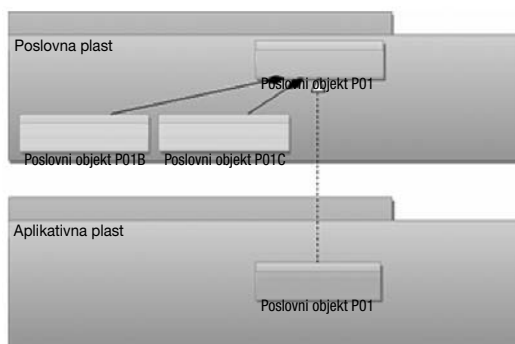
4.3.1 Vzorec aplikativno podprtih poslovnih objektov

Če je poslovni objekt aplikativno podprt, potem ga realizira (vsaj eden) podatkovni objekt. Aplikativno podprte poslovne objekte lahko opredelimo z množico:

$$[FABO] \downarrow AnBO = \{a \mid a \in AnBO \wedge ((\exists b \in DO : (b,a) \in Realization) \vee (\exists bo \in BO : (((bo, a) \in Composition) \vee (bo, a) Aggregation)) \wedge (\exists c \in DO : (c, bo) \in Realization)))\} \quad (FD9)$$

Slika 7 prikazuje primer pogleda realizacije poslovnih objektov PO1, PO1B in PO1C, ki so aplikativ-

no podprti (realizirani) s podatkovnim objektom PO1.



Slika 7: **Aplikativno podprt poslovni objekt**

4.3.2 Vzorec delno aplikativno podprtih poslovnih objektov

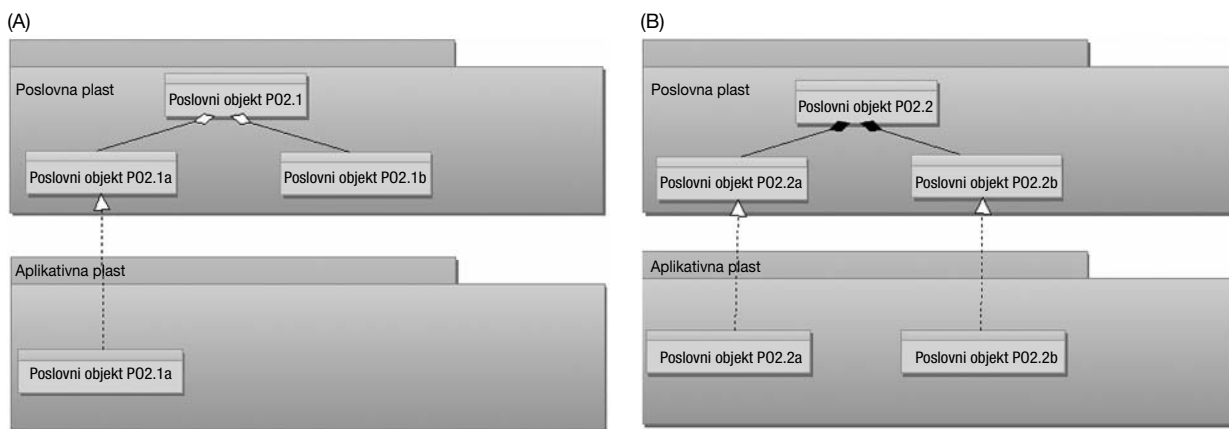
Poslovni objekt je delno aplikativno podprt, če ni aplikativno podprt v celoti (vzorec aplikativno podprtih poslovnih objektov) in je vsaj eden izmed njego-

vih delov realiziran s podatkovnim objektom. Poslovni objekt je lahko združuje več poslovnih objektov (agregacija) ali je sestavljen iz več poslovnih objektov (kompozicija). Delno aplikativno podprte poslovne objekte lahko opredelimo z množico:

$$[PABO] \downarrow AnBO = \{a \mid a \in AnBO \setminus [FABO] \downarrow AnBO \wedge (\exists b \in BO: (a,b) \in Composition \vee (a,b) \in Aggregation) \wedge (\exists c \in DO: (c,b) \in Realization)\}$$

(FD1)

Slika 8 prikazuje dva primera pogleda realizacije poslovnega objekta, ki kažeta na delno aplikativno podporo.



Slika 8: **Delno aplikativno podprt poslovni objekt**

4.3.3 Vzorec aplikativno nepodprtih poslovnih objektov

Poslovni objekt je aplikativno nepodprt, če niti poslovni objekt niti kateri izmed njegovih delov (drugi poslovni objekti, ki nastopajo v agregaciji ali kompo-

ziciji danega poslovnega objekta) ni realiziran s podatkovnim objektom. Aplikativno nepodprte poslovne objekte lahko opredelimo z množico:

$$[AUSBO] \downarrow AnBO = AnBO \setminus ((FABO - AnBO) \cup [PABO] \downarrow AnBO)$$

(FD2)

Slika 9 prikazuje primer pogleda realizacije poslovnega objekta PO3, ki je aplikativno nepodprt v poslovnem sistemu, vendar je realiziran z dokumentom D1.



Slika 9: **Aplikativno nepodprt poslovni objekt**

4.3.4 Vzorec neaplikativno podprtih poslovnih objektov in delno neaplikativno podprtih poslovnih objektov

Neaplikativno podprt poslovni objekt je realiziran z vsaj eno predstavitvijo. Neaplikativno podprte poslovne objekte lahko opredelimo z množico:

$$[FCBO]_{\downarrow} AnBO = \{a \mid a \in AnBO \wedge ((\exists b \in R: (b,a) \in Realization) \vee (\exists bo \in BO : (((bo, a) \in Composition) \vee (bo, a) \in Aggregation)) \wedge (\exists c \in R : (c, bo) \in Realization))))\} \quad (FD3)$$

Opazimo lahko, da je opredelitev vzorca neaplikativno podprtih poslovnih objektov analogna opredelitvi vzorca aplikativno podprtih poslovnih objektov (FD10). Prav tako je opredelitev vzorca delno neaplikativno podprtih poslovnih objektov analogna vzorcu delno aplikativno podprtih poslovnih objektov. Zato je ne bomo posebej navajali.

Slika 9 prikazuje primer pogleda realizacije poslovnega objekta PO3, ki je neaplikativno podprt v poslovnem sistemu.

4.3.5 Vzorec strogo nepodprtih poslovnih objektov

Primeri, ko poslovni objekt ne bi bil realiziran niti s predstavitvijo niti s podatkovnim objektom, so zelo

redki. To bi namreč pomenilo, da poslovni objekt v poslovnem sistemu ni kakor koli dokumentiran, temveč obstaja le kot tacitna informacija. Če obstajajo v opisu poslovno-informacijske arhitekture strogo nepodrti poslovni objekti, to največkrat pomeni, da arhitekturni opis ni pravilen. Koristno je, če orodje opozori uporabnika na to. V tem primeru lahko uporabnik model bodisi ustrezno popravi ali potrdi, da poslovni objekt res nima realizacije. Če poslovni objekt nima realizacije in hkrati nastopa v poslovnih procesih, pomeni, da gre za situacijo, ki bi jo bilo treba nasloviti.

Strogo nepodprte poslovne objekte lahko opredelimo z množico:

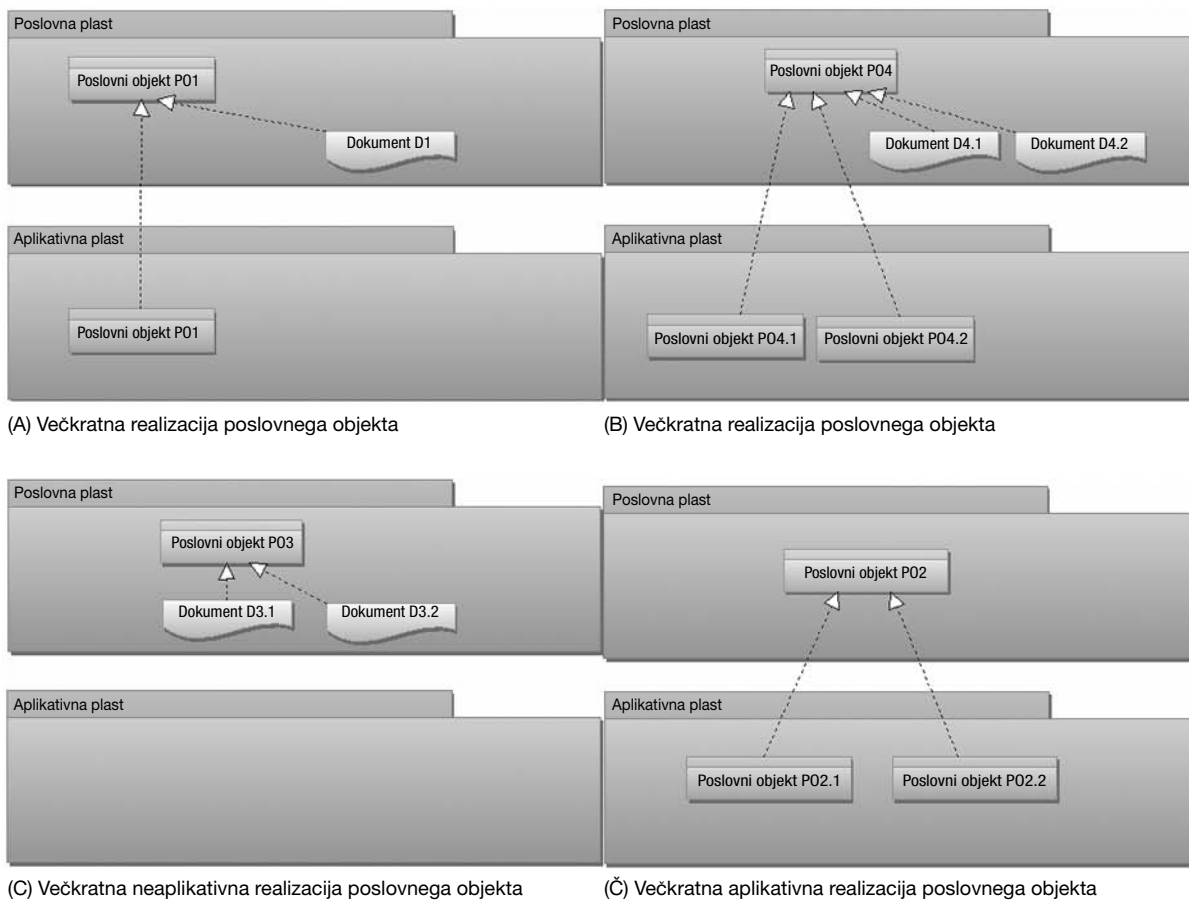
$$[URBO]_{\downarrow} AnBO = AnBO \setminus (, FCBO - AnBO. [\cup PCBO]_{\downarrow} AnBO \cup [FABO]_{\downarrow} AnBO [\cup PABO]_{\downarrow} AnBO) \quad (FD4)$$

Pri tem $PCBO_{AnBO}$ predstavlja množico delno neaplikativno podprtih poslovnih objektov.

4.3.6 Vzorec večkratne realizacije poslovnega objekta

Posamezni poslovni objekt je lahko v poslovnem sistemu večkrat realiziran, npr. z različnimi dokumenti

v papirni obliki, v več podatkovnih bazah itn. Vzorec večkratne realizacije poslovnega objekta največkrat kaže na neoptimalne strukture in na podvajanje podatkov v poslovnem sistemu, ki lahko vodi v znane probleme, kot sta npr. težje vzdrževanje podatkov in večje tveganje za nekonsistentnost podatkov.



Slika 10: **Večkratna realizacija poslovnega objekta**

Naj bo $BOR(bo)$ funkcija, ki dani poslovni objekt bo preslika v množico elementov, ki ga realizirajo:

$$BOR(bo) = \{rdo \mid (rdo \in R \cup DO) \wedge (((rdo, bo) \in Realization) \vee (\exists bo1 \in BO : ((bo1, bo) \in Composition) \vee ((bo1, bo) \in Aggregation)) \wedge (rdo, bo1) \in Realization))\} \quad (FD5)$$

Poslovne objekte, ki so v poslovnem sistemu večkrat realizirani, lahko opredelimo z množico:

$$[MRBO]_{\downarrow} AnBO = \{a \mid a \in AnBO \wedge |BOR(a)| \geq 2\} \quad (FD6)$$

Zgornje slike (slika 10 (A–Č)) prikazujejo različne primere večkratne realizacije poslovnega objekta.

4.3.7 Vzorec enkratne realizacije poslovnega objekta

Vzorec enkratne realizacije poslovnega objekta pomeni, da ima poslovni objekt v poslovnem sistemu natanko eno realizacijo. Ta je lahko bodisi aplikativna ali neaplikativna.

Poslovne objekte, ki so v poslovnem sistemu enkrat realizirani, lahko opredelimo z množico:

$$[SRBO]_{\downarrow} AnBO = \{a \mid a \in AnBO \wedge |BOR(a)| = 1\} \quad (FD7)$$

Primeri enkratne realizacije poslovnega objekta prikazujeta sliki 7 in 9.

5 SKLEP

V prispevku smo predstavili vlogo arhitekturne analize na področju poslovno-informacijskih arhitektur. Predstavili smo problematiko področja, ki izhaja iz pomanjkanja mehanizmov, ki bi omogočali iskanje določenih tipičnih struktur, pomembnih za analitika na učinkovit način. V ta namen smo predlagali uporabo vzorcev v poslovno-informacijski arhitekturi in opredelili množico vzorcev za ugotavljanje ravni podpore poslovnih procesov in poslovnih objektov. Predstavljeni analitski vzorci omogočajo zaznavanje tipičnih struktur v poslovno-informacijski arhitekturi in njihovo primerjavo. Vzorci so formalizirani na način, ki omogoča njihovo implementacijo in avtomatizacijo zaznavanja struktur. Ker je fokus na medplastnem povezovanju, predlagani analitski vzorci temeljijo na jeziku ArchiMate.

Organizacija *The Open Group* pripravlja novo različico jezika ArchiMate, ki bo integrirala pristopa ArchiMate in TOGAF ter ohranjala skladnost z obstoječima različicama pristopov. Integracija obeh pristopov bo predstavljala naslednjo razvojno stopnjo dveh najbolj priznanih pristopov, integriranih v enega samega. Predlagani analitski vzorci bodo uporabni tudi za naslednjo različico ogrodja.

Avtorji so predstavljene analitske vzorce uporabili pri strateškem planiranju informatike v petih velikih slovenskih poslovnih sistemih s področja elektrodistribucije, telekomunikacij, javne uprave in finančnega trgovanja. Izkazali so se kot koristno orodje pri analizi obstoječega stanja ter kot formalna podlaga za odločitve pri izdelavi strateškega načrta. Avtorji zato obstoječo različico strateškega planiranja enotne metodologije razvoja informacijskih sistemov [7] dopolnjujejo s pristopi poslovno-informacijskih arhitektur z uporabo predstavljenih vzorcev.

Ana Šaša je leta 2005 diplomirala in leta 2009 doktorirala na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Ukvarja se s poslovno-informacijskimi arhitekturami, avtomatizacijo poslovnih procesov, storitveno usmerjenimi arhitekturami, usklajevanjem poslovne domene in IT-domene, sistemi za podporo odločanju, ontologijami in strateškim planiranjem. Sodelovala je na številnih znanstvenoraziskovalnih projektih in projektih iz gospodarstva. Je članica več strokovnih in znanstvenih združenj. Objavila je vrsto prispevkov v priznanih mednarodnih revijah ter na domačih in tujih konferencah. Leta 2009 je prejela prvo nagrado za raziskovalne dosežke Fakultete za računalništvo in informatiko Univerze v Ljubljani.

Marjan Krisper je izredni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je tudi predstojnik katedre za informatiko in predstojnik laboratorija za informatiko. Njegova bibliografija obsega več kot dvesto strokovnih člankov in znanstvenih razprav. 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 for Information Systems) in član izvršnega odbora Slovenskega društva Informatika.

6 LITERATURA

- [1] Chief Information Officer Council (2001). A Practical Guide to Federal Enterprise Architecture. Dostopno na: <http://www.enterprise-architecture.info/Images/Documents/Federal%20Enterprise%20Architecture%20Guide%20v1a.pdf>.
- [2] Fowler, M. (1997). Analysis patterns: reusable object models. Addison-Wesley.
- [3] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2002). Design patterns: abstraction and reuse of object-oriented design. Software pioneers: contributions to software engineering. Springer-Verlag New York, Inc., 701–717.
- [4] IEEE Computer Society (2000). IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE Standard 1471-2000.
- [5] James, G. A., Handler, R. A., Lapkin, A., Gall, N. (2005). Gartner Enterprise Architecture Framework: Evolution 2005. Gartner.
- [6] Krisper, M. et al. (2003). EMRIS – Enotna metodologija razvoja informacijskih sistemov – Strateško planiranje. Ljubljana, Vlada Republike Slovenije, Center Vlade RS za informatiko.
- [7] Krisper, M., & Rožanec, A.: Obvladovanje informatike v poslovnih sistemih : pomen strategije in arhitektur. *Uporabna informatika*, 13(4), 185–198.
- [8] Lankhorst, M. et al. (2009). Enterprise Architecture at Work – Modelling, Communication and Analysis, 2. izdaja, Springer Berlin Heidelberg.
- [9] Riehle, D., & Züllighoven, H. (1996). Understanding and using patterns in software development. *Theor. Pract. Object Syst.*, 2, 3–13.
- [10] Rožanec, A., & Krisper, M. (2009). Poslovno informacijska arhitektura uprave. V *Povežimo informacijske otoke: zbornik konferenčne Informatika v javni upravi*. Kongresni center Brdo pri Kranju, Ljubljana, Slovensko društvo Informatika.
- [11] The Open Group (2009). ArchiMate 1.0 Specification, Van Haren Publishing.
- [12] The Open Group (2010a). The ArchiMate Forum. Dostopno na: <http://www.opengroup.org/archimate/>.
- [13] The Open Group (2010b). The Power of Enterprise Architecture. Dostopno na: <http://www.archimate.org>.
- [14] The Open Group (2008). TOGAF™ Version 9, TOGAF Series, Van Haren Publishing, 9. izdaja.
- [15] Zachman J. A. (1987). A Framework for Information System Architecture, *IBM System Journal*, 26(3), 276–292.
- [16] Zachman, J. A. (2010). The Zachman Framework™, <http://www.zachmaninternational.com/>.