



Creative Commons Attribution –
NonCommercial 4.0 International License

Prethodno priopćenje

<https://doi.org/10.31784/zvr.11.1.7>

Datum primitka rada: 15. 12. 2022.

Datum prihvaćanja rada: 15. 2. 2023.

MODEL ODRŽAVANJA POSLOVNIH APLIKACIJA U UVJETIMA PROMJENE POSLOVNOG OKRUŽENJA

Miodrag Sretenović

Dr. sc., programer, MI&DA d.o.o., Šulekova 29, 47000 Karlovac, Hrvatska; e-mail: m.sretenovic@mida.hr

Božidar Kovačić

Dr. sc., izvanredni profesor, Fakultet informatike i digitalnih tehnologija Sveučilišta u Rijeci,
Radmile Matejčić 2, 51000 Rijeka, Hrvatska; e-mail: bkovacic@inf.uniri.hr

SAŽETAK

Odgovor na promjene poslovnog okruženja zahtjeva modifikaciju i dodavanje novih funkcionalnih komponenti unutar poslovnih aplikacija. Prioritet održavanja je zadržavanje integriteta podatka i strukture svih validiranih izvještaja. Izmjenom arhitekture sustava generirane karakteristike poslovne aplikacije moraju sadržavati funkcionalne zahtjeve korisnika i zahtjeve promjene regulatora višeg informacijskog sustava na koji je niži informacijski sustav oslonjen. Opisani model održavanja poslovnih aplikacija definiran je funkcijom prijelaza iz početnog u konačno stanje automata uz uvjet zadržavanja konzistentnosti podataka i izvještaja uvođenjem vremenske komponente za dohvat podataka i izvještaja. Funkcija prijelaza primjenom vremenskom komponentom rezultira usmjerenim grafom stanja koji se sastoji od vremenskih čvorova koji prikazuju stanje automata, a završno stanje prikazuje stanje konačnog automata. U radu se prikazuje stanje automata dobivenog primjenom vremenske komponente modela održavanja poslovnih aplikacija u uvjetima izmjene Zakonske fiskalne politike, intervencijom na Poreznu politiku dodavanje nove porezne stope.

Ključne riječi: Vremenska komponenta, transformacijski model, održavanje poslovnih aplikacija, održavanje poreznih aplikacija

1. UVOD

Izazov informatizaciji u sve većoj trendovskoj globalizaciji postavljaju sve teže zahtjeve kako prema krajnjem korisniku tako i prema regulatoru. Nepobitne činjenice su da su u svijetu velike količine podataka, strateška prednost onoga koji može iz velike količine informacija filtrirati korisne podatke, te na temelju tako dobivenih podataka donijeti brzu odluku. Sa druge strane zahtjevi regulatora višeg informacijskog sustava prvenstveno su usmjereni na unapređenje funkcionalnosti te na sigurnost svojih izloženih sustava. Izloženost sustava manifestira se prvenstveno zbog razmjene

informacija, koja se odvija dvosmjerno i to u smjeru nižeg informacijskog sustava prema većem i obrnuto. Razmjena podataka uobičajeno se vrši putem datoteka formata (pdf, xml, rtf, txt, csv, xls, xlsx, doc, docx i drugih datoteka). Nerijetko regulatori mijenjaju strukturalnu shemu datoteka za razmjenu podataka, a integracija i važnost kompatibilnosti informacijskih sustava sve više postaje bitan faktor u poslovnom svijetu pa se niži informacijski sustavi moraju prilagoditi novonastalim uvjetima.

Primjena standarda u fazi razvoja i arhitekture rezultira dobivanjem maksimalne karakteristike softvera.

Metodološkom primjenom standarda ISO/IEC 25010:2011 omogućavamo:

- funkcionalnost
- pouzdanost
- prikladnost za uporabu
- učinkovitost
- prikladnost za održavanje
- prenosivost
- kompatibilnost
- sigurnost (International Organization for Standardization ISO, 2011).

Prikladnost softvera za održavanje jedan je od ISO faktora ocjene kvalitete. Unaprjeđenjem kvalitete softvera u procesu razvoja smanjuje se intenzitet aktivnosti na održavanju softvera. Implementiranje svih funkcionalnosti sustava tijekom razvoja softvera od iznimne je važnosti jer implementiranje funkcionalnosti u procesu održavanja rezultiranje lošijim performansama softvera (Sretenović i Kovačić, 2020).

Unatoč dosljednoj primjeni standarda ISO/IEC 25010:2011, nije moguće u potpunosti predvidjeti sve aspekte održavanja, ali je nužno održavanje temeljiti na dosljednim metodološkim planskim principima održavanja poslovnih aplikacija.

Očekivani rezultati primjenom modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja su:

- definiranje modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja
- integracija modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja u poslovne aplikacije
- izgradnja prototipa modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja
- verifikacija funkcionalnosti modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja u stvarnom okruženju.

Kako bi korisnik dobio pravovaljane i brze odgovore, informacijski sustavi se moraju redovito prilagođavati, unapređivati i održavati.

Automatizacija zahtijeva određenu dozu jasnoće procesa već u fazi projektiranja, a modeliranje procesa kao i sam razvoj aplikacije moraju pratiti suvremene trendove s krajnjim ciljem zadovoljstva korisnika (Sretenovic *et al.*, 2016).

Definirana su sljedeća istraživačka pitanja:

1. Model održavanja baziran na vremenskoj komponenti primjenjiv je u procesima održavanja poslovnih aplikacija?
2. Predloženim modelom održavanja baziran na vremenskoj komponenti smanjuje vrijeme održavanja poslovnih aplikacija?

Istraživanju i koncipiranju rada pristupili smo metodološki principom autora Gregora i Hevnera kroz šest poglavlja i to:

- Uvod
- Pregled literature
- Metode
- Opis artefakta
- Evaluacija i rasprava
- Zaključak (Gregor i Hevner, 2013).

2. PREGLED LITERATURE

Informacijski sustavi izloženi su vanjskim utjecajem i možemo ih prezentirati zahjevima korisnika i funkcionalnim okruženjem. Troškovi održavanja su veliki zbog same činjenice da je životni vijek softvera vezan za održavanje. Nakon predaje softvera u eksploatacijsku produkciju pa do njegovog gašenja, period je kada je održavanje neophodno, kako bi zadovoljio funkcionalnim zahjevima korisnika, tako i funkcionalnim zahjevima viših informacijskih sustava regulatora. Nakon što je softver ušao u fazu eksploatacije bilo kakva intervencija na softveru smatra se održavanje.

IEEE [IEEE610.12-1990] je standard koji definira sposobnost programskog inženjerstva za održavanje softvera radi zadovoljenja specficiranih zahtjeva [ISO, 1990]. Ovim standardom definirane su kategorije kvaliteta održavanja s ciljem umanjenja troška održavanja, te pristupa održavanju.

Sistemske pristupom održavanja određujemo podjelu održavanja na: održavanje sheme ili pogleda te održavanje usmjereno na verzioniranje shema.

U svome istraživanju, skup formalnih operacija održavanja autori Blaschka *et al.* baziraju na formalnom modelu uređene šestorke te navode da je poslovno okruženje posljedica korisnikovog zahtjeva na održavanje, u svom radu model održavanja pristupaju modelom održavanja sheme.

Formalni model definiraju kao:

$$M = (Fc, Ld, Ab, Gr, Cl, At)$$

gdje su konačni skupovi:

Fc – činjenica

Ld – dimenzija

Ab – atributa

Funkcije:

Gr – povezivanja činjenica i dimenzija

Cl – relacije

At – povezivanja atributa činjenice i razine dimenzija.

U okviru definiranog modela autori su definirali skup formalnih operacija održavanja. Konceptualnim pristupom model definira održavanje sheme, te nakon funkcionalne prilagodbe slijedi proces implementacije (Blaschka *et al.*, 1999).

Rad autora Papastefanatos *et al.* iz područja Održavanje verzioniranjem sheme održavanje definira rješavanjem problema analizom podataka na izvoru. U svojem istraživanju predlaže održavanja jedinstvenim modelom relacije te modelom grafa I prikazom stanja, pogleda i upita (Papastefanatos *et al.*, 2007).

Rad na temu održavanja pogleda autori Bouzeghoub i Kedad opisuju procesom postupnog održavanja pogleda, te u svojem istraživanju navode da se proces mora sageldavati u četiri prizme dimenzija i to: informacije, jezika, promjena i instance (Bouzeghoub i Kedad, 2000).

Autor Djerdjouri u svom radu pojašnjava ključnu ulogu umjetne inteligencije u konkurentnosti malih i srednjih poduzeća te potrebu što je većeg opsega ulaganja i implementacije aplikativnih rješenja baziranih na umjetnoj inteligenciji. Nadalje navodi da su bez poslovne inteligencije tvrtke izložene riziku donošenja kritičnih odluka na temelju bilo nedovoljne ili netočne informacija. Korištenjem umjetne inteligencije omogućava se jednostavno kreiranje detaljnih financijskih izvješća, izvješća o operacijama, korisnicima i dobavljačima (Djerdjouri, 2020).

Autor Waszkowski u svom radu pojašnjava nove trendove u razvoju i održavanju poslovnih aplikacija. Platforma niskog koda omogućuje brzo generiranje i isporuku poslovnih aplikacija s minimalnim naporom pisanja programskog koda a ujedno zahtijeva najmanji mogući napor za instalaciju i konfiguraciju okruženja, te obuka i implementacija. Ovaj rad opisuje korištenje niskokodne platforme Aurea BPM za automatizaciju poslovanja (Waszkowski, 2019).

Autori Rahman *et al.* u svom radu akcentiraju vremensku komponentu održavanja, navode da održavanje aplikacije oduzima znatnu količinu vremena i sredstava svake godine. Gotovo 60 % IT proračuna troši se samo na održavanje aplikacija. Razloge za korištenjem usluga održavanja vanjskih korisnika nije samo u smanjenje troškova održavanja, već i oslobađanje resurse i zadržanju fokusa na temeljnim proizvodima. Korištenjem usluga održavanja vanjskih korisnika je uobičajena poslovna strategija koje tvrtke koriste za postizanje uštede troškova od oko 20 - 50 %. Međutim, proces donošenja odluka održavanja aplikacije složena je pojava. Temelji se na nizu čimbenika utjecaja, klijentovih zahtjeve i prirodu projekta (Rahman *et al.*, 2020).

3. METODE

Studija Metodologija istraživanja znanosti o dizajnu (DSRM) je metodologija koju koristimo u ovom istraživačkom radu. Ova metodologija istraživanja znanosti o dizajnu provodi se u šest koraka, a to su:

- identifikacija problema i motivacija
- definiranje ciljeva za rješenje
- dizajn i razvoj
- demonstracija
- evaluacija
- rasprava (Hevner *et al.*, 2004).

4. OPIS ARTEFAKTA

4.1 Identifikacija problema i motivacija

Porezno zakonodavstvo je jedno od najdinamičnijih zakonodavstava. Globalni trendovi govore o čestim intervencijama fiskalnih vlasti na porezne sustave svojih država. Svaka intervencija na fiskalni sustav za sobom povlači niz promjena na obveznim Zakonskim sadržajima obrazaca izvješćivanja. Porezno Zakonodavstvo propisuje obvezne forme, rokove i načine dostave podataka, te rokove čuvanja poslovne dokumentacije koji su u rasponu od nekoliko godina pa sve do obveze trajnog čuvanja pojedinih porezno-financijskih izvještaja.

Poslovni informacijski sustavi kako bi imali svoju funkcionalnost obvezni su pratiti zakonodavni trend, uz uvjet zadržavanja integriteta svih prethodnih podataka o stanju, obliku i sadržaju izvještaja.

Financijska izvješća su završni cilj cjelokupnog računovodstvenog sustava (Žager *et al.*, 2008).

Financijski izvještaji nastaju kao proces sakupljanja podataka relevantnih za područje izvješćivanja. Cilj obrade i procesiranje podataka je sastavljanje i prezentiranje računovodstvenih informacija.

Identifikacija problema temelji se na prilagodbi informacijskog sustava uvjetovanoj zahtjevima regulatora. Zakonom o izmjenama i dopunama Zakona o porezu na dodanu vrijednost (NN br. 113/2022) uvodi se nova porezna stopu u Porezni sustav (nulta stopa PDV-a 0 %) na isporuku i ugradnju solarnih ploča na privatne stambene objekte te javne i druge zgrade koje se koriste za aktivnosti od javnog interesa te isporuku i ugradnju solarnih ploča u blizini takvih objekata, prostora i zgrada.

Rezultat Zakonske odluke o izmjenama i dopunama Zakona o porezu na dodanu vrijednost (NN br. 113/2022), je donošenje Pravilnika o izmjenama i dopunama pravilnika o porezu na dodanu vrijednost (NN, 133/2022) koji predviđa novi oblik i sadržaj financijskih obrazaca: Obrazac PDV, Knjige izdanih računa (I-RA) i Knjige primljenih računa (U-RA).

Promjene u informacijskom sustavu za potrebe uvođenja nulte stope PDV-a su:

- Dodavanje-izmjena strukture tablica relacijske baze uzrokovanu dodavanjem nulte stope PDV-a.
- Dodavanje i izmjena forme uzrokovana dodavanjem nulte stope PDV-a.
- Dodavanje i izmjena izvještaja uzrokovana dodavanjem nulte stope PDV-a.
- Testiranje i implementacije modula uzrokovana dodavanjem nulte stope PDV-a.

Izvor podataka Tablice 1, aktivnosti modela i kolone prosječno vrijeme u minutama rezultat je istraživanja i mjerenja dobivena online anketnim upitnikom provedenih pomoću programa Google Forms iz doktorskog rada Sretenović (Sretenovic, 2019). Provedeno istraživanje izvršeno je na reprezentativnom uzorku kod pravnih subjekata koji su razvrstani po Nacionalnoj klasifikaciji djelatnosti područje J - Informacije i komunikacije sukladno Odluci o Nacionalnoj klasifikaciji djelatnosti 2007 - NKD 2007 (NN br. 58/2007).

Kako se radi o održavanju poslovnih aplikacija iz domene promjene u okviru PDV-a, dobivene rezultate iz istraživanje modela aktivnosti i kolone prosječno vrijeme u minutama možemo koristiti u procesu istraživanja, dok su podaci o broju akcija izmjereni u procesu održavanja.

Tablica 1. Ukupno prosječno očekivano održavanja

Rb	Aktivnosti modela	Prosječno vrijeme (min)	Broj akcija	Ukupno
1.	<i>Promjena strukture relacijskih baza podataka</i>			
	– dodavanje atributa u relacijsku tablicu	1,50	15	22,50
	– dodjele domene i svojstava dodanim atributima	1,00	15	15,00
	– aktivacija atributa	0,50	15	7,50
	– dodavanje generatora	2,00	2	4,00
	– aktivacija generatora	0,50	2	1,00
Utrošeno vrijeme na dodavanje-izmjena strukture tablica relacijske baze uzrokovanu dodavanjem nulte stope PDV-a.				50,00
2.	<i>Dodavanje nove forme</i>			
	– promjena vizualnog izgleda konfiguriranih objekata	30,50	2	61,00
	– dodavanje akcije na dodanim atributima unutar forme.	2,00	20	40,00
Utrošeno vrijeme na dodavanje-izmjenu forme uzrokovanu dodavanjem nulte stope PDV-a.				101,00

3.	<p>Proces dodavanja novog upita $Q_{n,m+1}$ u relacijskoj bazi podataka $RLDT_n$</p> <p>Svojstva novog upita $m+1$ definiraju se funkcijom za dodjelu svojstava u okviru domene upita:</p> <ul style="list-style-type: none"> – definiranje $P_{n,m+1}$ – projekcije (II), na inicijalnu tablicu $RTB_{n,t}$ u relacijskoj bazi $RLDT_n$ – definiranje $PR_{n,m+1}$ – skupa tablica (\otimes), za izvođenje upita $Q_{n,m+1}$ u relacijskoj bazi $RLDT_n$ – definiranje $V_{n,m+1}$ – veze između tablica \bowtie, veze relacijske inicijalne tablice $RTB_{n,t}$ upita i tablice spajanja $RTB_{n,u}$, za element skupa atributa – definiranje $F_{n,m+1}$ – formula (σ), formula s logičkim izrazom za izdvajanje n-torki iz produkta tablica $P_{n,m+1}$ upita $Q_{n,m+1}$ – aktivacija upita. 	8,50	3	25,50
		8,50	3	25,50
		12,50	3	37,50
		5,00	3	15,00
		0,50	3	1,50
Proces dodavanja novog upita $Q_{n,m+1}$ u relacijskoj bazi podataka $RLDT_n$				105,00
4	<p>Proces dodavanja novog izvještaja $REP_{n,m+1}$ u relacijsku bazu podataka $RLDT_n$</p> <p>Svojstva novog izvještaja $m+1$ definiraju se funkcijom za dodjelu svojstava u okviru domene izvještaja:</p> <ul style="list-style-type: none"> – definiranje $P_{n,m+1}$ – projekcije (II), na inicijalnu tablicu $RTB_{n,t}$ u relacijskoj bazi $RLDT_n$ – definiranje $PR_{n,m+1}$ – skup tablica (\otimes), za izvođenje izvještaja $Q_{n,m+1}$ u relacijskoj bazi $RLDT_n$ – definiranje $V_{n,m+1}$ – veze između tablica \bowtie, veze relacijske inicijalne tablice $RTB_{n,t}$ izvještaja i tablice spajanja $RTB_{n,u}$, za element skupa atributa – definiranje $F_{n,m+1}$ – formula (σ), formula sa logičkim izrazom za izdvajanje n-torki iz produkta tablica $P_{n,m+1}$ izvještaja $Q_{n,m+1}$. – definiranje $RED_{n,m+1}$ – definiranje redoslijeda pojavljivanja vrijednosti u izvještaju, – definiranje $PRIZ_{n,m+1}$ – definirana svojstva izvještaja (naslov izvještaja, sažetak izvještaja, zaglavlje izvještaja, podnožje izvještaja, glavno zaglavlje, glavno podnožje, detaljni podaci, grupno zaglavlje, grupno podnožje, format izvještaja). – aktivacija izvještaja 	7,00	3	21,00
		7,00	3	21,00
		12,50	3	37,50
		6,00	3	18,00
		4,00	3	12,00
		7,00		7,00
		0,50		0,50
Utrošeno vrijeme na dodavanje-izmjene izvještaja uzrokovanu dodavanjem nulte stope PDV-a.				109,50
5	Testiranje i implementacije	40,00	1	40,00
Utrošeno vrijeme na testiranje i implementacije modula uzrokovanu dodavanjem nulte stope PDV-a				40,00
Ukupno prosječno očekivano vrijeme (u minutama):				405,50

Izvor: autor i doktorski rad Sretenović (2019)

Očekivano prosječno vrijeme održavanja poslovnih aplikacija temeljem Zakonske promjene uvođenja nulte stope PDV-a, iznosila bi 405,50 minuta.

Pokazatelji u tablici 1. daju legitimitet, motivaciju i opravdanosti projektiranja arhitekture modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja i primjenu vremenske komponente u održavanje. Ciljevi održavanju jesu izbjegavanje intervencije u programski kod, a teret i odgovornost održavanja prenijeti na korisnika i administratora informacijskog sustava korisnika što upravo daje predloženi model.

4. 2 Definiranje ciljeva za rješenje

Razvojem modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja u optimalizira se proces poslovanja u uvjetima digitalnih transformacija.

Skraćivanjem vremena potrebnog za izvršenje procesa održavanje povećavamo produktivnost a u konačnici povećavamo kvalitetu isporučenog softvera te zadovoljstvo korisnika.

Prijenosom obveze održavanja sa projektanta i servisnih timova na administratora korisnikovog sustava koji je u kompetenciji korisnika, oslobađaju se resursi softverskih tvrtki prema procesu daljnjeg istraživanja te projektiranju novih softverskih proizvoda. Sukladno tome zadovoljstva izvršenom uslugom te optimalizacijom i kvalitetom usluge nalazi se i na strani isporučitelja, projektanta softverskog rješenja.

Modelom održavanja definirat ćemo komponente sustava održavanja i njihove međusobne odnose baziranog na vremenskoj komponenti i definiranim čvorovima stanja.

Rezultat je transformacija relacijske baze podataka i njezinih komponenti te stanja formi, upita i izvještaja uvjetovanih promjenom stanja vremenska komponente.

4. 3 Dizajn i razvoj

Dizajn modela održavanja uključuje:

1. Definiranje formalnog zapis transformacijskog modela održavanja
 - a. definiranje formalnog zapisa strukture relacijske baze podatka (RLDT)
 - b. uvođenje vremenske komponente transformacijskog modela i definiranje algoritama transformacijskog modela za prijelaze između stanja RLDT
2. izrada konceptualnog modela održavanja

Dijagram prijelaza između stanja RLDT uvođenjem vremenske komponente modela predstavlja usmjereni graf. Čvorovi grafa odgovaraju stanjima automata a grane prijelaza označene su alfabetom. Prikazom funkcije prijelaza između čvorova dobivamo konačni automat (Slika 1.).

Formalni model održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja zasnovan je na relacijskoj algebri i teoriji skupova.

Formalno možemo definirati uređenom petorkom, gdje je konačni skup stanja definiran:

$$MR = (K, p, \alpha, q_0, ZS)$$

Za koji vrijedi da je:

$K = MR = \{F_1, F_2, F_3, \dots, F_n\}$ – konačni skup stanja

$p = \{0, 1\}$ - iskazna logika

α – funkcija prijelaza, za koju vrijedi da je

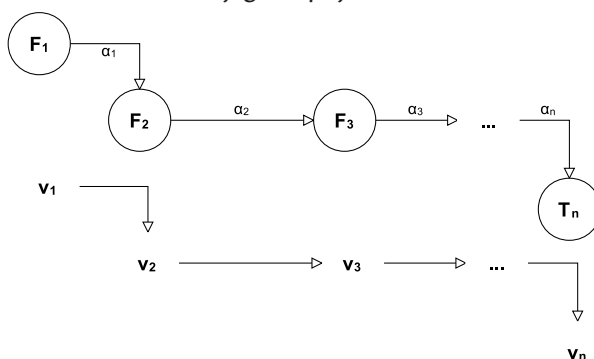
$\alpha: K \times p \rightarrow R(K)$ gdje je $R(K)$ particija od K ,

dok je stanje prijelaza definirano: $F_1 \rightarrow F_2, F_2 \rightarrow F_3, \dots, F_{n-1} \rightarrow F_n$

q_0 – početno stanje, za koju vrijedi da je $q_0 \in K, F_1$ za α_1 vrijeme implementacije i predstavlja početno stanje

ZS – skup završnih stanja, $ZS \subseteq K$, opisuje je stanje F_n za trenutak v_n .

Slika 1. Dijagram prijelaza modela



Izvor: Autor

Formalni zapis transformacijskog modela uključuje formalni zapis strukture baze podataka te uvođenje vremenske komponente transformacijskog modela te definiranjem algoritama transformacijskog modela. Zbog detaljnosti formalnog zapisa transformacijskog modela, prezentiran je samo formalni zapis transformacije RLDT, tablica i atributa.

Relacijskih shema za relacijsku bazu definiramo je skupom:

$RSHEMA_1 = \{ZSHE_1, ZSHE_2, \dots, ZSHE_n\};$

$ZSHE_1 = \{nosh_1, vrem_1, vrem_2\}$ – zapis relacijske sheme modela za datumski period

$nosh_1$ – identifikator modela za relacijsku bazu

$vrem_1$ – datum početka primjene

$vrem_2$ – datum kraja primjene

Element strukture relacijske baze modela:

$$ES_{TRLDTn,i} = (RLDT_n, (nosh_i, \{0, 1\})); n, i \in N$$

$RLDT_n$ – relacijska baza podataka

$nosh_i$ – identifikator modela

Funkcija aktivacije i deaktivacije transformacijskog modela prikazana je u tablici 2.

Tablica 1. Tablica prijelaza modela za relacijsku bazu

Staro stanje	Funkcija aktivacije	Novo stanje
$ES_{TRLDTn,i} = (RLDT_n, (nosh_i, 0))$	$f_1(RLDT_n, nosh_i) \rightarrow t_s: 0 \rightarrow 1$	$ES_{TRLDTn,i} = (RLDT_n, (nosh_i, 1))$
Staro stanje	Funkcija deaktivacije	Novo stanje
$ES_{TRLDTn,i} = (RLDT_n, (nosh_i, 1))$	$f_2(RLDT_n, nosh_i) \rightarrow t_s: 1 \rightarrow 0$	$ES_{TRLDTn,i} = (RLDT_n, (nosh_i, 0))$

Izvor: autor

Element strukture podatka tablice $RTB_{n,t}$ relacijske baze $RLDT_n$ čini uređeni par:

$$ES_{TBn,t} = (RLDT_n, RTB_{n,t}); n, t \in N$$

$RLDT_n$ – relacijska baza podataka

$RTB_{n,t}$ – relacijska tablica u relacijskoj bazi $RLDT_n$.

Funkcija aktivacije i deaktivacije transformacijskog modela prikazana je u tablici 3.

Tablica 2. Tablica prijelaza transformacijskog modela za relacijske tablice.

Staro stanje	Funkcija aktivacije	Novo stanje
$ES_{TRTBn,i} = (ES_{TBn,t}, (nosh_i, 0))$	$f_3(ES_{TBn,t}, nosh_i) \rightarrow t_s: 0 \rightarrow 1$	$ES_{TRTBn,i} = (ES_{TBn,t}, (nosh_i, 1))$
Staro stanje	Funkcija deaktivacije	Novo stanje
$ES_{TRTBn,i} = (ES_{TBn,t}, (nosh_i, 1))$	$f_4(ES_{TBn,t}, nosh_i) \rightarrow t_s: 1 \rightarrow 0$	$ES_{TRTBn,i} = (ES_{TBn,t}, (nosh_i, 0))$

Izvor: autor

Element strukture podatka atributa $at_{n,t,m}$ tablice $RTB_{n,t}$ relacijske baze $RLDT_n$ čini uređena trojka:

$$ES_{ATn,t,m} = (RLDT_n, RTB_{n,t}, at_{n,t,m}); n, t, m \in N$$

$RLDT_n$ – relacijska baza podataka

$RTB_{n,t}$ – relacijska tablica u relacijskoj bazi $RLDT_n$

$at_{n,t,m}$ – atribut relacijske tablice $RTB_{n,t}$ relacijske baze $RLDT_n$.

Funkcija aktivacije i deaktivacije transformacijskog modela prikazana je u tablici 4.

Tablica 3. Tablica prijelaza modela za attribute relacijske tablice

Staro stanje	Funkcija aktivacije	Novo stanje
$ES_{TRATn,t,m} = (ES_{ATn,t,m}, (nosh_i, 0))$	$f_5(ES_{ATn,t,m}, nosh_i) \rightarrow t_s: 0 \rightarrow 1$	$ES_{TRATn,t,m} = (ES_{ATn,t,m}, (nosh_i, 1))$
Staro stanje	Funkcija deaktivacije	Novo stanje
$ES_{TRATn,t,m} = (ES_{ATn,t,m}, (nosh_i, 1))$	$f_6(ES_{ATn,t,m}, nosh_i) \rightarrow t_s: 1 \rightarrow 0$	$ES_{TRATn,t,m} = (ES_{ATn,t,m}, (nosh_i, 0))$

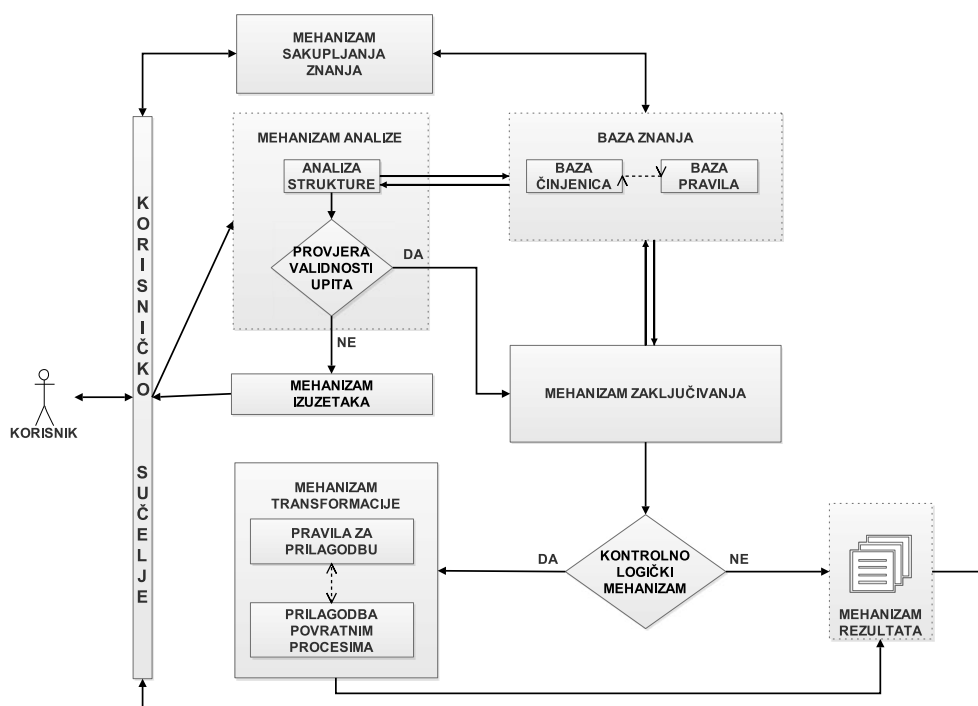
Izvor: autor

Na slici 2 prikazan je konceptualni model korišten za razvoj informacijskog sustava podržanog za modelom održavanja zasnovanog na vremenskoj komponenti.

Elementi modela prikazani su na slici 2 i sastoje se od:

- mehanizam za sakupljanje znanja - u procesu prikupljanja znanja vrše se procesi analize, provjera i oblikovanja koje pohranjujemo u bazu znanja
- baza znanja - sastoji se od dva modula i to baza činjenica u kojem se evidentiraju početna stanja i baza pravila u kojem se nalaze operatori čija je funkcija pretvaranja stanja problema u rješenje
- mehanizam analize, ima ulogu provjeravati upite te pronaći podatke u bazi činjenica za datumski period
- mehanizam izuzetaka ima ulogu da vrati korisniku informaciju o definiranju upita izvan datumskog perioda
- mehanizam zaključivanja, ima ulogu usporedbe činjenica sa pravilima, te na temelju toga zaključuje o pokretanju akcije koje će se izvršiti prema pravilima u bazi pravila, za upit prosljeđen od mehanizma analize pretraživanjem baze znanja donosi odluku o pokretanju akcija
- kontrolno-logički mehanizam, nastavlja se na mehanizam zaključivanja, u slučaju da je upit izvan datumskih kriterija u mehanizmu rezultata neće biti promjena u odnosu na trenutno stanje sustava, dok u uvjetima da je upit unutar datumskih varijabli putem mehanizma transformacije pokreće se prilagodba postojećeg stanja novom stanju sustava.
- mehanizam transformacije, sastoji se od pravila za prilagodbu i prilagodbe povratnim procesima te nakon potvrde kontrolno-logičkog mehanizma nakon prilagodbi dolazi do mehanizma rezultata
- mehanizam rezultata, mehanizam u kojem se rezultiraju prethodni koraci
- korisničko sučelje, omogućuje interakciju korisnika sa sustavom.

Slika 2. Konceptualni model



Izvor: autor

4.4 Demonstracija

Tendencija razvoja aplikacija je zasnovana na temeljima multiplatformskog pristupa, stoga se u razvoju modela održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja pristupilo takvom principu. Razvoj multiplatformskih aplikacija u osnovi je duži u razvoju, ali projekt koji je po svojim komercijalnim ciljevima ima veći broj mogućih kupaca.

Stoga je imperativ izbora alata za razvoj modela bio vođen imperativom multiplatformne aplikacije. Programsko rješenje model održavanja poslovnih aplikacija u uvjetima promjene poslovnog okruženja izrađeno je sa Embarcadero RAD studio 10.3 Rio alatima Delphi 10.3 Rio i C++ Builder 10.3 Rio. Korištena je ultrabrz, skalabilna SQL baza podataka InterBase XE7.

Programski kod dodavanje atributa demonstriran je u Tablici 5, dok je u Tablici 6 prikazan programski kod prijelaza modela za atribut relacijske tablice.

Tablica 5. Programski kod dodavanja atributa

Programski kod
<pre> IBQ.SQL.Add('SELECT r.RDB\$FIELD_NAME AS fd_nm,'); IBQ.SQL.Add('r.RDB\$FIELD_ID AS fd_ID,'); IBQ.SQL.Add('r.RDB\$DESCRIPTION AS fd_desc,'); IBQ.SQL.Add('r.RDB\$DEFAULT_VALUE AS fd_def_value,'); IBQ.SQL.Add('r.RDB\$NULL_FLAG AS fd_not_null_constraint,'); IBQ.SQL.Add('f.RDB\$FIELD_LENGTH AS fd_length,'); IBQ.SQL.Add('f.RDB\$FIELD_PRECISION AS fd_precision,'); IBQ.SQL.Add('f.RDB\$FIELD_SCALE*(-1) AS fd_scale,'); IBQ.SQL.Add('CASE f.RDB\$FIELD_TYPE'); IBQ.SQL.Add('WHEN 7 THEN + "SMALLINT"); IBQ.SQL.Add('WHEN 261 THEN + "BLOB"); IBQ.SQL.Add('WHEN 14 THEN + "CHAR"); IBQ.SQL.Add('WHEN 40 THEN + "CSTRING"); IBQ.SQL.Add('WHEN 11 THEN + "D_FLOAT"); IBQ.SQL.Add('WHEN 27 THEN + "DOUBLE"); IBQ.SQL.Add('WHEN 10 THEN + "FLOAT"); IBQ.SQL.Add('WHEN 16 THEN + "DECIMAL"); IBQ.SQL.Add('WHEN 8 THEN + "INTEGER"); IBQ.SQL.Add('WHEN 9 THEN + "QUAD"); IBQ.SQL.Add('WHEN 12 THEN + "DATE"); IBQ.SQL.Add('WHEN 13 THEN + "TIME"); IBQ.SQL.Add('WHEN 35 THEN + "TIMESTAMP"); IBQ.SQL.Add('WHEN 37 THEN + "VARCHAR"); IBQ.SQL.Add('ELSE + "UNKNOWN"); IBQ.SQL.Add('END AS fd_type,'); IBQ.SQL.Add('f.RDB\$FIELD_SUB_TYPE AS fd_subtype'); IBQ.SQL.Add('FROM RDB\$RELATION_FIELDS r'); IBQ.SQL.Add('LEFT JOIN RDB\$FIELDS f ON r.RDB\$FIELD_SOURCE = f.RDB\$FIELD_NAME'); IBQ.SQL.Add('WHERE r.RDB\$RELATION_NAME=:tblime'); IBQ.SQL.Add('ORDER BY RDB\$FIELD_ID'); IBQ.Params[0].value := IBQ1RDBRELATION_NAME.Value; </pre>

Izvor: autor

Tablica 6. Tablica prijelaza modela za atributa relacijske tablice r.RDB\$RELATION_NAME

Staro stanje	Novo stanje
f.RDB\$FIELD_NAME_AC.Value := 0	f.RDB\$FIELD_NAME_AC.Value:= 1
Staro stanje	Novo stanje
f.RDB\$FIELD_NAME_AC.Value:= 1	f.RDB\$FIELD_NAME_AC.Value:= 0

Izvor: autor

Implementacijom programskim rješenjem dokazujemo da je model održavanja baziran na vremenskoj komponenti primjenjiv u procesima održavanja poslovnih aplikacija. U procesu

održavanja predloženi model održavanja oslonjiv je na jednu bazu podataka, ujedno arhitektura modela omogućava da baza podataka sadrži sve potrebne podatke za transformaciju relacijske baze tijekom vremena. Što čini prednost u odnosu na tradicionalni oblik održavanja koji nalaže obveznu intervenciju na programskom kodu, koja u konačnici nalaže kreiranje nove izvršne verzije poslovne aplikacije i novu bazu podataka po izvršenoj promjeni.

5. EVALUACIJA I RASPRAVA

Učinkovitost modela demonstrirana je usporedbom vremena potrebnog u procesu održavanja uslijed Zakonskih izmjena uvođenjem nulte stope PDV-a.

Provedenom procjenom zaključujemo da je moguće implementirati i dizajnirati model koji je ubrzao proces održavanja poslovnih aplikacija, čime smo odgovorili na drugo postavljeno istraživačko pitanje. Doprinos ovog istraživanja je model koji ima cilj unaprijediti poslovanje u segmentu poslovnih aplikacija, te unaprijediti održavanje i životni ciklus aplikacija.

Održavanje poslovnih aplikacija iz područja poreznog Zakonodavstva je posebno osjetljiv dio poslovanja zbog niza faktora koji određuje Zakonodavac u svojim pravilima. Svaka pogreška može uzrokovati Zakonske posljedice za korisnika. Korištenjem predloženog modela povećavamo produktivnost i zadovoljstvo korisnika. Ukupni doprinos ovog istraživanja sastoji se u primjeni modela u rješavanju problema automatizacije procesa održavanja poslovnih aplikacija bez intervencije u programski kod aplikacija. Problem održavanja poslovnih aplikacija vrlo su česti, a cilj ovog istraživanja je smanjiti vrijeme i utjecaj intervencije na programski kod. Ujedno ovakvi modeli održavanja predstavljaju automatizirane procese i pomažu korisniku u uštedi vremena, novca i resursa. Ovaj model automatiziranog sustava ubrzava poslovne procese iz domene menadžmenta i iz domene primjene i pridržavanja Zakonskih pravila. Ovakvi automatizirani modeli umanjuju mogućnost pogreške, prvenstveno zbog minimalnog ljudskog utjecaja na proces. Budućnost nam sigurno donosi nove promjene u shemi vezanoj za porezno zakonodavstvo, te ostavlja prostor za daljnje unapređenje modela. Buduća istraživanja trebaju biti usmjerena na potpunu automatizaciju poslovnih procesa održavanja uz što manje intervencije na strukturnu promjenu programskog koda i oblika te strukturnu promjenu sheme baze podataka.

Tablica 7. Rezultati potrebnog vremena za održavanje poslovnih aplikacija

Rb	Anketno pitanje	Aktivnosti modela	Prosječno vrijeme održavanja		
			Bez Modela	Sa Modelom	Razlika
1.	Utrošeno vrijeme na dodavanje-izmjena strukture tablica relacijske baze uzrokovanu dodavanjem nulte stope PDV-a	Promjena strukture relacijskih baza podataka (dodavanje dvaju atributa).	50,00	32,88	17,12
2.	Utrošeno vrijeme na dodavanje-izmjenu forme uzrokovanu dodavanjem nulte stope PDV-a.	Promjena postojeće forme.	101,00	65,45	35,55

3.	Proces dodavanja novog upita $Q_{n,m+1}$ u relacijskoj bazi podataka $RLDT_n$	Promjena upita.	105,00	50,40	54,60
4.	Utrošeno vrijeme na dodavanje-izmjene izvještaja uzrokovanu dodavanjem nulte stope PDV-a	Promjena izvještaja.	109,50	60,28	49,22
5.	Utrošeno vrijeme na testiranje i implementacije modula uzrokovanu dodavanjem nulte stope PDV-a	Testiranje i implementacije	40,00	25,00	15,00
Ukupno prosječno očekivano vrijeme (u minutama):			405,50	234,01	171,49

Izvor: autor i doktorski rad Sretenović (2019)

Izvor podataka Tablice 7 kolona prosječno vrijeme održavanja provedenih pomoću programa Google Forms iz doktorskog rada Sretenović (Sretenovic, 2019), bez primjene modela i primjenom modela na reprezentativnom uzorku kod pravnih subjekata koji su razvrstan po Nacionalnoj klasifikaciji djelatnosti u područje J - Informacije i komunikacije, nedvojbeno potvrđuje opravdanost projektiranja i primjene modela. Iz same tablice razvidno je da je očekivano prosječno vrijeme održavanja poslovnih aplikacija bez primjene modela 405,50 minuta, dok je očekivano vrijeme održavanja primjenom modela 234,01 minuta. Prosječna očekivana razlika iznosi 171,49 minuta.

6. ZAKLJUČAK

Pouzdana i brzo dostupna usluga održavanja danas je tendencija u poslovnim procesima. Za očekivati je da će se porezno-financijski standardi nadograđivati i unapređivati, kako na prijedloge korisnika, teko i na prijedloge Zakonodavca. Predloženi model može se primijeniti i na specijalizirane izvještaje korisnika, brzina informacija u poslovnom svijetu je ponekad presudna, tako da predloženi model ima još veću težinu u svojoj primjeni.

Radi se o modelu kojeg je moguće implementirati i koji ima jedinstven pristup održavanju, ima niz direktnih i indirektnih prednosti. Direktna prednost su brzina, lakoća održavanja te izbjegavanje intervencije u programski kod, nadalje omogućava jedinstven pristup izradi izvještaja po zahtjevu korisnika. Sa druge strane ovakav model štedi novac i financijske resurse potrebne za održavanje, kao i ne manje vrijednu činjenicu da ovakav model produljuje životni vijek poslovne aplikacije. Krajnji cilj modela je pomoći nositeljima donošenja odluka o korištenju modela, kao i ponuditi smjernice za razvoj vlastitih modela održavanja baziranog na ovom modelu.

LITERATURA

- Blaschka, M., Sapia, C., Höfling, G. (1999, August). On schema evolution in multidimensional databases. In International Conference on Data Warehousing and Knowledge Discovery (pp. 153-164). Springer, Berlin, Heidelberg.
- Bouzeghoub, M., Kedad, Z. (2000, September). A logical model for data warehouse design and evolution. In International Conference on Data Warehousing and Knowledge Discovery (pp. 178-188). Springer, Berlin, Heidelberg.

- Djerdjouri, M. (2020). Data and Business Intelligence Systems for Competitive Advantage: prospects, challenges, and real-world applications. *Mercados y Negocios*, (41), 5-18.
- Gregor, S., Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS quarterly*, 337-355.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.
- ISO/IEC 25010:2011, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, International Standards Organization for Standardization, Ženeva, Švicarska, 2011.
- ISO/IEC IEEE610.12-1990 IEEE Standard Glossary of Software Engineering Terminology, International Standards Organization for Standardization, Ženeva, Švicarska,
- Odluka o Nacionalnoj klasifikaciji djelatnosti 2007. - NKD 2007 (2007) Narodne novine broj 58/2007
- Papastefanatos, G., Vassiliadis, P., Simitsis, A., & Vassiliou, Y. (2007, September). What-if analysis for data warehouse evolution. In *International Conference on Data Warehousing and Knowledge Discovery* (pp. 23-33). Springer, Berlin, Heidelberg.
- Pravilnik o izmjenama i dopunama pravilnika o porezu na dodanu vrijednost (2022), Narodne novine broj 133/2022
- Rahman, H. U., Raza, M., Afsar, P., Khan, H. U., & Nazir, S. (2020). Analyzing factors that influence offshore outsourcing decision of application maintenance. *IEEE Access*, 8, 183913-183926.
- Sretenovic, M., Kovacic, B. (2020). Model payment order in the SEPA system. *International Journal of Business Information Systems*, 33(4), 531-548.
- Sretenović, M. (2019). Adaptivni transformacijski model održavanja poslovnih aplikacija (Doctoral dissertation, University of Rijeka. Department of Informatics).
- Sretenović, M., Kovačić, B., Jovanović, V. (2016, May). Organization of tax data warehouse for legal entities. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1482-1487). IEEE.
- Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, 52(10), 376-381.
- Zakon o izmjenama i dopunama Zakona o porezu na dodanu vrijednost (2022), Narodne novine broj 113/2022
- Žager K., Mamić Sačer I., Sever S., Žager L. (2008), Analiza financijskih izvještaja, Masmedia



Creative Commons Attribution –
NonCommercial 4.0 International License

Preliminary communication

<https://doi.org/10.31784/zvr.11.1.7>

Received: 15. 12. 2022.

Accepted: 15. 2. 2023.

MODEL OF MAINTENANCE OF BUSINESS APPLICATIONS IN THE CONDITIONS OF CHANGING BUSINESS ENVIRONMENT

Miodrag Sretenović

PhD, developer, MI&DA d.o.o., Šulekova 29, 47000 Karlovac, Croatia; e-mail: m.sretenovic@mida.hr

Božidar Kovačić

PhD, Associate Professor, Faculty of Informatics and Digital Technologies, University of Rijeka, Radmile Matejčić 2, 51000 Rijeka, Croatia; e-mail: bkovic@inf.uniri.hr

ABSTRACT

Responding to changes in the business environment requires modification and addition of new functional components within business applications. The maintenance priority is maintaining the integrity of the data and the structure of all validated reports. By changing the system architecture, the generated characteristics of the business application must contain the functional requirements of the user and the requirements of changing the regulator of the higher information system on which the lower information system is based. The described business application maintenance model is defined by the function of transition from the initial to the final state of the automaton with the condition of maintaining the consistency of data and reports by introducing a time component for retrieving data and reports. The transition function applied with a time component results in a directed state graph consisting of time nodes representing the state of the automaton, and the final state representing the state of the final automaton. The paper shows the state of the automaton obtained by applying the time component of the business application maintenance model in the conditions of changing the Legal Fiscal Policy, adding a new tax rate through the intervention of the Tax Policy.

Key words: Time component, transformation model, maintenance of business applications, maintenance of tax applications

