

# Transformacijski model programskog koda knjigovodstvenih aplikacija u uvjetima promjene poreznih stopa

Miodrag Sretenović

MI&DA d.o.o., Bogoslava Šuleka 29, Karlovac, Hrvatska

e-mail: [m.sretenovic@mida.hr](mailto:m.sretenovic@mida.hr)

**Sažetak** - Modeliranje i razvoj nekog informacijskog sustava može se smatrati uspješnim jedino ako u kratkom roku informacijski sustav daje odgovore korisniku na sve korisničke zahtjeve. Modeliranje i razvoj knjigovodstvenih aplikacija mora sadržavati odgovore na zahtjev korisnika o obliku i sadržaju svih minimalnih Zakonskih evidencija, knjiga i izvještaja za praćenje poreznih evidencija. U radu je opisan transformacijski model programskog koda primijenjen na knjigovodstvene aplikacije.

Tijekom višegodišnje analiza poslovnih procesa u evidenciji i bilježenju poslovnih promjena u Zakonom predviđenim evidencijama, knjigama i izvještajima razvijen je model prilagodbe programskog koda poslovnim promjenama. Definirani su funkcionalni zahtjevi i arhitektura informacijskog sustava izvještaja u programskom kodu u Zakonom predviđenih evidencija, knjiga i izvještaja, te je utvrđeno da je postojeći model razvoja i održavanja knjigovodstvenih aplikacija moguće unaprijediti.

**Ključne riječi:** transformacijski model, poslovna aplikacija, porezne aplikacija, paralelno programiranje, generator, generativno programiranje.

## 1. UVOD

Informacijski sustavi poduzeća za praćenje poreznih evidencija spadaju u grupu informacijskog sustava uredskog poslovanja, sustav uredskog poslovanja je element informacijskog sustava koji daje podršku u smislu manipulacije poreznim dokumentima.

Predmet istraživanja je identifikacija i razrješenje problema produljenja poslovne funkcionalnosti i životnog ciklusa uredskih aplikacija za praćenje poreznih evidencija, a u uvjetima Zakonskih promjena poreznih stopa.

Cilj ovoga rada je ukazati na probleme održavanja poslovnih aplikacija u uvjetima promjene Zakonom propisanih poreznih propisa, te prikazati model razvoja koji bi ovaj segment poslovnih aplikacija poboljšao i digao na jednu veću razinu profesionalnosti.

Dugoročni trendovi u poreznim politikama nam govore o čestim poreznim promjenama, takvi trendovi poreznih politika su direktno međusobno povezani sa IT sektorom,

na način da knjigovodstvene aplikacije u svojim modulima izvješćavanja moraju pratiti sve Zakonom predviđene porezne promjene, što za posljedicu ima česte strukturalne promjene koda i formi aplikacija kao i strukturalne promjene elemenata baze podataka.

U današnje vrijeme u uporabi se češće koriste termini "Paralelno računarstvo" (*Parallel computing*) i "Računarstvo visokih performansi" (*High Performance Computing*) i super-računarstvo (*Supercomputing*), od pojma paralelno programiranje, i to ne bez razloga. U vremenu potreba za brzom i točnom informacijom sve je teže naći načine i modele dolaska do brze i točne informacije. Tu na scenu stupaju paralelni računarski sustavi te paralelni softveri. Općenito paralelnog računarstva je forma u kojoj se veliki broj operacija obavlja istovremeno. [1,2,3].

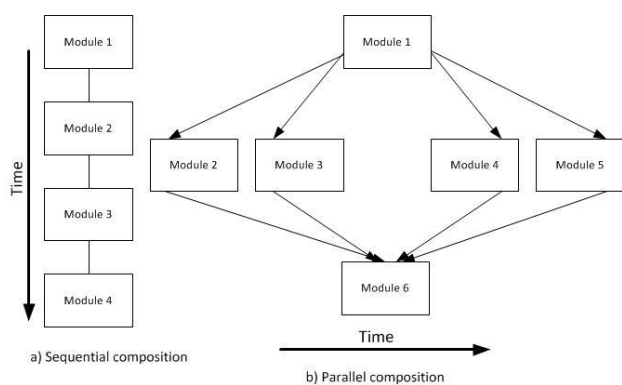
Paralelno programiranje je posebna grana programiranja koja proučava metode oblikovanja, programskog ostvarivanja i izvršavanja računalnih programa gdje se koriste posebne metode za razdvajanje algoritama ili problema na njegove osnovne dijelove i izvode se istovremeno tj. paralelno.

U nastavku se opisuje integracija dviju danas aktualnih tehnologija na području programiranja, generativnog programiranja i jezika skripata na način da je izrađen koncept generativnog programiranja temeljen na jezicima skripata [4]. Naglasak je stavljen na izradu modeliranja generatora aplikacija, te na generativnom razvoju, kao postupku paralelnog razvoja aplikacija i generatora. Time se postiže skraćanje razvojnog ciklusa aplikacija, optimizacija performansi, te pojednostavljenje održavanja.

U poglavlju 2 opisan je koncept paralelnog modela, poglavlje 3 prezentira uporabljivost modela, a nakon toga u poglavlju 4 prikazani su rezultati istraživanja primjene metoda u svijetu, u poglavlju 5 prikazuje se transformacijski model programskog koda za praćenje poreznih evidencija, nakon čega u poglavlju 6 slijedi prikaz stanja u RH i inozemstvu i u poglavlju 7 slijedi zaključak i predviđanje daljnjeg razvoja na ovom području.

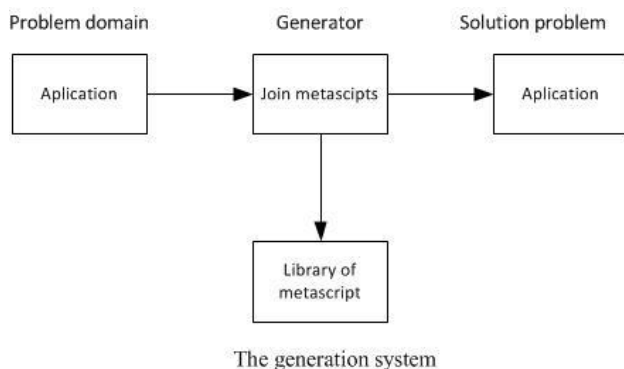
## 2. KONCEPT PARALELNOG PROGRAMIRANJA

Kako bi se program mogao istodobno izvršavati mora biti paraleliziran, odnosno podijeljen na dijelove koji se mogu izvršavati nezavisno jedni od drugih. Slika 1 prikazuje primjer računalnog programa koji se sastoji od više osnovnih programskih blokova. Slika 1a. prikazuje slijedno izvođenje programa u kojem se pojedini blokovi izvršavaju slijedno jedan iza drugog. Slika 1b prikazuje paralelno izvođenje programa u kojem se programski blokovi 2, 3, 4, 5 izvršavaju istodobno [5,6].



Slika 1. Modeli programiranja

Generativno programiranje je disciplina automatskog programiranja koji je započeta pod tim imenom, u kasnim 90-im godinama prošlog stoljeća. Prema ovoj definiciji, generativno programiranje predstavlja "... osmišljavanje i provedba programskih modula koji se mogu kombinirati za stvaranje specijaliziranih i visoko optimizirane sustave koji ispunjavaju određene uvjete", Eisenecker [7]. Težnja za programiranje kod optimizacije čini, prema [7], glavni specifične razlike prema drugim tehnikama automatskog programiranja.

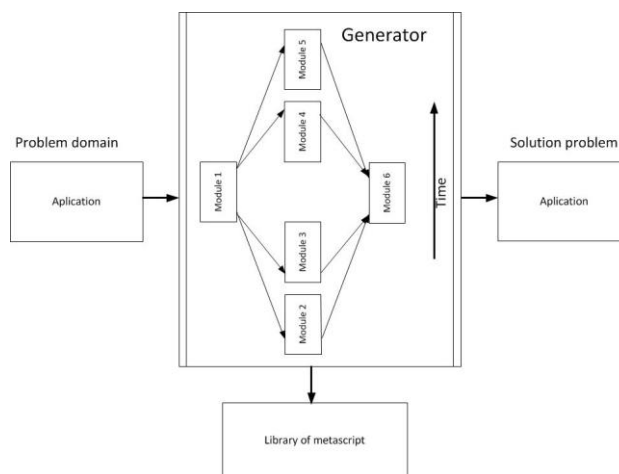


Slika 2. Osnovni model generatora

Slika 2 prikazuje osnovni model generatora prema Eisenecker [7], što govori da ovakvi modeli imaju visoki

stupanj apstrakcije, i zadaću unapređenja kompleksnosti aplikacija. A kao fundamentalnu uslugu imaju povezivanje komponenti u jedinstvenu cjelinu informacijskog sustava. Zbog svoje apstrakcije, automatizirano izvršavaju programirane transformacije unutar aplikacije.

Skriptni model generatora aplikacija razvijen je za potrebe generativnog programiranja. Za razliku od objektnog modela, zadaća skripti je zadana unutar problemske domene, a na sve funkcionalnosti aplikacije, pošto su skripte zadane na nižoj razini, tj. u predlošcima koda aplikacije, a koji se u okviru skriptnog modela nazivaju metaskriptama.



Slika 3. Transformacijski skriptni model generatora

Na slici 3 prikazan je transformacijski skriptni model generatora, gdje je unutar skriptnog modela generatora prikazana apstraktivnost zadane problemske domene.

Na slici 3 u bloku generator tako razvijeni generatori mogu se upotrijebiti u okviru novih projekata, ali se također mogu ugrađivati i u stare projekte uz pridržavanje uputa modeliranja takvih blok generatora. Kod starih projekata, potrebno je izvršiti pretvorbu slijednog u paralelni algoritam, apstrahiranjem poreznog segmenta aplikacije i njegovim paralelnim razvojem u uvjetima promjene poreznih stopa, što je i prikazano na slici 3.

Ovakvi modeli projektiranja informacijskih sustava imaju svoju težinu znajući da je samo Zakon o porezu na dodanu vrijednost u primjeni u više od 150 zemalja, a svoju popularnost duguju svojoj izdašnosti punjenja proračuna. Uslijed globalne krize i sve veće financijske potrebe državnih proračuna, on izravno utječe i na njegove česte izmjene, a samim tim i globalni način izvještavanja poreznih tijela domicilnih zemalja [8].

Kao posljedica obvezujuća je korektivno-adaptivno održavanje aplikacija, kako bi se zadovoljile Zakonske norme i pravila, i kako bi se produljio životni vijek validnosti aplikacije uzrokovanih Zakonskim izmjenama.

To je u praksi značilo da moduli informacijskih sustava koji imaju zadaću praćenje poreznih propisa na propisanim obrascima, a koji su dio cjeline

više-komponentnog informacijskog sustava, prolaze kroz korektivnu izmjenu.

Izmjena uključuje izmjenu:

- programskog koda,
- korektivno-adaptivnu izmjenu strukture baze,
- izmjenu postojećih formi unosa podataka,
- izrada novih formi unosa i pregleda podataka.

Kao rezultat gore navedenih činjenica za posljedicu su nastajali česti zastoji u radu te veću nematerijalnu štetu neizdavanjem propisanih dokumenta i neposjedovanje propisanih obrazaca. Ujedno zbog korektivno-adaptivne aktivnosti na programskom kodu aplikacija, neplanirane pogreške i zastoje nisu rijetki, te su često aplikacije završavale u stanju u potpunog zastoju. (engl. deadlock).

### 3. PRIJAŠNJA ISTRAŽIVANJA

Upotreba metoda i modela odavno je prepoznata kao dobra praksa tokom procesa razvoja informacijskih sustava [9]. Korištenje modela u procesu razvoja informacijskih sustava prisutno je već desetljećima. Pojavom metodologija i alata koji omogućavaju kreiranje softverskih modela na visoko apstraktnom nivou.

Pristup koji podrazumijeva upotrebu modela kao ključne i nerazdvojive komponente razvojnog procesa, postao je jedan od vodećih pristupa u razvoju softverskih sustava [10, 11, 12].

Razvoj poreznih aplikacija možemo podijeliti u nekoliko faza: analiza elemenata softvera, specifikacija, arhitektura, implementacija, testiranje, izrada dokumentacije i održavanje [13, 14, 15, 16].

Razvoj poreznih aplikacija zahtjeva visoku apstraktnost aplikacija, pošto su vrlo zahtjevni i kao za cilj imaju strogo zadane forme koje su propisane Zakonskim aktima koji propisuju minimalne parametre.

Sama analiza elemenata softvera i specifikacija je faza u kojoj se precizno definiraju karakteristike koje softver mora zadovoljavati.

U praksi je većina kvalitetnih specifikacija napisana tako da isprofilira već gotove parametre i varijable, sukladno Zakonskim potrebama izvješćivanja, dok se arhitektura softverskog sustava za praćenje poreznih evidencija odnosi na apstraktnu reprezentaciju tog sustava. Arhitektura se brine da softverski sustav zadovolji zahtjeve proizvoda, ali isto tako osigura da budući zahtjevi mogu biti riješeni nadogradnjom.

Arhitektura također mora omogućavati povezivanje između softverskog sustava i ostalih softverskih modula u poslovnim aplikacijama. Modul za praćenje poreznih evidencija tek je jedan od segmenta šireg poslovno aplikativnog rješenja.

Testiranje je važan dio izrade kvalitetnog softvera kojim se pokušava uočiti i ukloniti greške koje su nastale prije implementacije, te općenito optimiziraju softverski proizvod. U praksi bi to značilo da li poslovna aplikacija za praćenje poreznih evidencija zadovoljava prvenstveno

Zakonske okvire izvješćivanja, sa svim propisanim elementima i parametrima, i tek nakon toga da li zadovoljava zahtjeve klijenta.

Izrada dokumentacije poslovnih aplikacija, a samim tim i poslovnih aplikacija za praćenje poreznih evidencija u svrhu budućeg održavanja i nadogradnje je u praksi vrlo često zanemaren proces. Dizajneri i programeri vrlo često zanemaruju potrebu izrade pripadajućih uputa smatrajući da je ovaj itekako važan element manje važan u razvoju i implementaciji poreznih i poslovnih aplikacija. Naravno da dokumentacija mora postojati, mora biti pisana jezgrovito, jasno i precizno kako bi se maksimalno olakšao posao budućeg održavanja, i prvenstveno kako bi se olakšalo korištenje poreznih aplikacija od strane korisnika.

Praksa nam govori da je segment podrške u osposobljavanju na korištenju poreznih aplikacija itekako zanemaren kao jedan od najbitnijih čimbenika u razvoju i implementaciji aplikacija.

Razlog velikog postotka neuspjelih poslovno aplikativnih projekata je u tome što razvojni timovi ne uspiju naći dodatne resurse na kvalitetnom održavanju poslovnih aplikacija.

Oko 70% informacijskih sustava u svijetu se smatra neuspješnim. U prosjeku se troškovi i rokovi za izradu sustava prekorače za oko duplo, a oko trećine projekata se prekine prije dovršenja [17, 18]. Loše planiranje ili projektiranje novog sustava gotovo će sigurno rezultirati neuspjehom.

Loše ili djelomično održavanje u periodu nakon implementacije sigurno će završiti neuspjehom projekta, pa je to ujedno i razlog da je pri izradi informacijskih sustava potrebno pristupiti s inženjerskog stajališta.

### 4. ISTRAŽIVANJA

U razvoju porezno-poslovnih aplikacija možemo koristiti slijedne (bez uporabe generatora) ili paralelne modele razvoja porezno-poslovnih aplikacija s upotrebom generatora [19].

Razvoj poslovno-poreznih aplikacija bez uporabe generatora je uobičajena metoda razvoja poslovnih aplikacija, te po svojoj strukturi pripada razvoju softvera kao niz slijednih transformacija, te je apstrahiran trenutno važećim Zakonskim propisima i možemo ga kvalificirati kao informacijski sustav i model zatvorenog tipa.

Zbog svoje zatvorenosti ima niz manjkavosti i nedostataka te postoje osnovane pretpostavke i razlozi za neuspjeh slijednih softverskih projekata za praćenje poreznih evidencija bez uporabe generatora [20], a to su:

- Loša izvedba projekta,
- Loše upravljanje izmjenama u uvjetima promjene poreznih stopa,
- Zanemarivanje potrebe za izmjenama,
- Prekoračenje zadanih rokova,
- Nerealno planiranje rokova,
- Loše upravljanje zahtjevima,

- Loša ili nepostojeća specifikacija zahtjeva,

Možemo reći da je područje paralelno generativnog programiranja često zastupljeno u suvremenim istraživanjima. Primjena paralelno-generativnih aplikacija zastupljeno je u velikom broju znanstvenih i apstraktnih informacijskih sustava.

Autori BARABÁSI, Albert-László; ALBERT, Réka. pozivaju se na primjenu paralelno generativnog algoritma u segmentu genetske medicine i primjeni modela Barabasi-Albertova algoritma [21].

Na primjeru živih sustava koji čine veliku genetsku mrežu, a čiji vrhovi su proteini i geni, rubovi predstavljaju kemijske interakcije između njih. Na drugoj organizacijskoj razini, velika mreža formirana od strane živčanog sustava, čiji vrhovi su živčane stanice, povezane nitima, neiscrpno je područje u istraživanjima i primjeni paralelno generativnog programiranja.

U ovom radu je dat i primjer paralelno generativnog algoritma u društvenim znanostima, gdje su vrhovi pojedinci ili organizacije, a rubovi obilježavaju socijalne interakcije između njih, također predstavlja neiscrpnu znanstvenu temu u istraživanju društveno-socioloških aspekata društvenih istraživanja.

Također je opisano funkcioniranje World Wide Weba (WWW), čiji vrhovi su HTML dokumenti povezani linkovima, koji upućuju iz jedne stranice u drugu. Zbog svoje velike veličine i složenosti interakcije, topologija tih mreža je u velikoj mjeri nepoznanica [21].

Autori Gomez-Cabrero, D., Menche, J., Cano, I., Abugessaisa, I., Huertas-Migueláñez, M., Tenyi, A., ... & Tegnér, J. u svom radu primjenjuju model Barabasi-Albertov model algoritma primijenjen u molekularno-medicinskom znanstvenom istraživanju pronalaska novih metoda, lijekova i biomarkera kod medicinske dijagnostike. U radu se ističe važnost istraživanja primjenom Barabasi-Albertovog modela algoritma [22].

Autori ALAM, Maksudul; KHAN, Maleq, u svom radu govore o uporabi paralelnih algoritama uz primjenu generatora u medicinskoj dijagnostici, sa ciljem stvaranja velike mreže slučajnih mreža. Koristeći algoritam s mogućnošću generiranja mreže od jedne milijarde čvorova i 250 milijardi niti, što bi u dijagnostici donijelo revolucionarni napredak [23].

Autori DOBRE, Ciprian; XHAFA, Fatos u svom radu govore o tendenciji obrade podataka na velikim bazama podataka, koja su sve više javno dostupne putem suvremenih informacijskih tehnologija. Obilje dostupnih podataka u širokom rasponu diže nove izazove i prilike u mnoštvo disciplina, u rasponu od znanosti i inženjerstva biologije i poslovanja. Jedan od glavnih izazova je kako

iskoristiti neviđenu razmjernu podataka, kako bi stekli daljnji uvid i znanje za poboljšanje kvalitete ponuđenih usluga.

U ovom radu analizirane su mogućnosti i izazovi za učinkovito paralelno procesiranje podataka. Big Data u Cloud okruženju kao jedna od suvremenih metoda istraživanja, te povećanje produktivnosti temeljem dobivenih rezultata.

U radu se utvrđuje potreba i učinkovitost paralelnih algoritama i način sinkronizacije Big Data u Cloud sustavu [24].

Autori RADOŠEVIĆ, Danijel; MAGDALENIĆ, Ivan u radu prikazuju model izvornog koda generatora na temelju dinamičkih okvira. Model se zove SCT model i sastoji se od tri osnovne komponente: Specifikacija (S), koja opisuje karakteristike aplikacija, konfiguracije (C), koji opisuje pravila za primjenu, te predlošci (T) [25].

Autori SMITH, Adam M.; MATEAS, Michael u radu govore o arhitekturi, dizajnu i primjeni generatora i procedura u modeliranju i izradi informatičkih igara, gdje je posebno akcentirana uloga generatora, te utvrđeno da je tržišni uspjeh proizvoda direktno povezan sa modeliranjem i arhitekturom generatora. Na temelju ulaznih parametara generator procedure generira izlazne segmente igara što direktno utječe na atraktivnost i popularnost igara, a samim tim i uspješnost proizvoda na tržištu [26].

Na temelju gore navedenih radova vidi se tendencija i perspektivnost, kako trenutnog tako i budućeg područja generativno paralelnog programiranja. U vremenu kada je brzina dolaska do informacija jedna od najbitnijih kako znanstvenih tako i praktičnih vrijednosti, budući utjecaj ovog segmenta praktičnog programiranja imate sve više na svojoj primjeni i značaju. Posebno značenje je uočeno na big data sustavima, koji putem generatora i paralelnih algoritama daju neograničenu kombinaciju mogućnosti i brzinu dolaska do traženih rezultata.

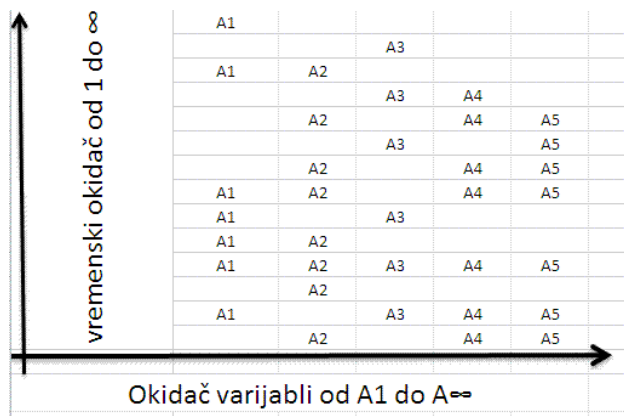
## 5. TRANSFORMACIJSKI MODEL PROGRAMSKOG KODA ZA PRAĆENJE POREZNIH EVIDENCIJA

Izrada razvojne okoline za paralelno programiranje zasnovana na tabličnom programatoru kao jedan od najraširenijih oblika programiranja bliskih krajnjim korisnicima u posljednjih se nekoliko godina nametnula tehnika tabličnog programiranja. Osim primjene za obavljanje složenih matematičkih izračuna, tablično programiranje je u posljednje vrijeme prepoznato i kao pogodna tehnika za razvoj raspodjeljenih računalnih sustava zasnovanih na globalnoj mreži Internet.

Temeljem ove činjenice možemo reći da je modeliranje generatora za potrebe poreznih evidencija zasnovanog na tehnici tabličnog programiranja u uvjetima promjena poreznih stopa najbliži korisniku po načinu korištenja metodologiji i brzini transformacije poreznih aplikacija.

Ovaj model je puno otporniji na pogreške, a model transformacije je automatski upravljani tabličnim programatorom. Razvoj tablične aplikacije bio je jedan od glavnih razloga koji su utjecali na opće prihvaćanje računala za poslovnu i osobnu upotrebu. Lakoća uporabe te odlično vizualno kreiranje složene logike. Tablično programiranje donosi istu tu učinkovitost u razvoj programa. U tabličnom programiranju se koriste tablične aplikacije poput Microsoft Excela kako bi se ostvarili i testirali algoritmi odlučivanja, simulacije, izvještaji ili bilo kakva druga vrsta složenih izračuna i procesuiranja podataka. Temeljem ovih činjenica izrađen je model generatora za programsko rješenje poreznih aplikacija u uvjetima izmjene poreznih stopa i to na način da smo primijenili sustav koordinatne osi uz dvije varijable:

- datumska varijabla,
- varijabla poreznih stopa.



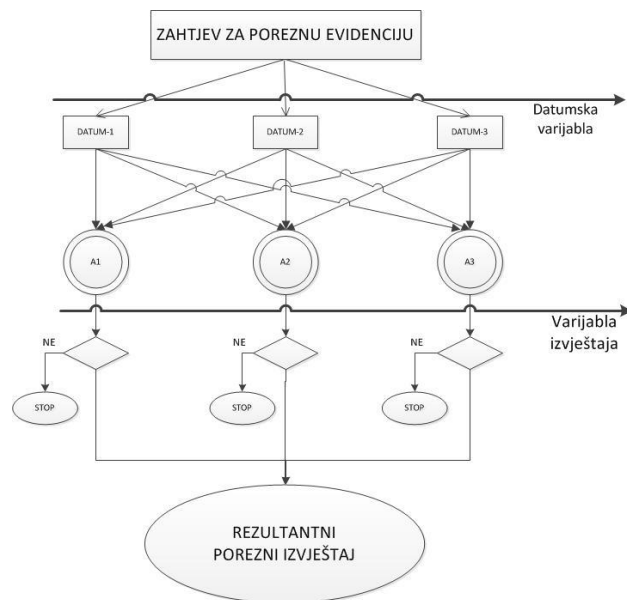
Slika 4 Model okidača generatora

Uloga ovako postavljenog okidača je da sukladno zahtjevu korisnika, a koji u okviru datumskog upita prikaže samo mrežom određene varijable.

Broj varijabli je u granicama od 1 do neograničeno puno, broj datumskih varijabli je također neograničeno velik.

Model čvorova [27] poznat je i kao Barabasi-Albertov model algoritma za generiranje neograničenih mreža koristi mehanizam preferencijalnog povezivanja slika 4. Na slici 5 prikazan je tok i načini povezivanja varijabli transformacijskog modela programskog koda.

Model čvorova u konkretnom primjeru započinje inicijalnim upitom korisnika za određeni datumski period dok je uloga okidača, a što je vidljivo i iz slike 5 da pokrene točno određene varijable, te prikaže kao rezultat traženi zahtjev što bi značilo temeljem slikovnog prikaza 5, da će pojedine varijable biti pokrenute, dok pojedine varijable ne.



Slika 5. Transformacijski model generatora

Paralelna obrada je poznata i kao postupak kod kojega se više instrukcija i varijabli obrađuje istovremeno [28]. Obrada se zasniva na principu da se veliki problemi gotovo uvijek mogu podijeliti na manje te onda obraditi istovremeno (paralelno). Više instrukcionalna obrada poreznih varijabli u traženom datumskom stanju, uz obradu svake porezne varijable koja odgovara traženom stanju datuma (slika 5), gdje je prikazana više instrukcionalna obrada varijable A, a koja čine elemente poreznih izvještaja u traženoj datumskoj varijabli, čime dobijemo porezno stanje dana kao uvjeta. Paralelne računalne programe je mnogo teže razvijati nego slijedne [9] zato što istodobnost uvodi nekoliko novih klasa potencijalnih mogućih grešaka od kojih je međusobno isključivanje, sinkronizacija i paralelno usporavanje najčešći poznatiji kao (eng. race condition). Komunikacija i sinkronizacija između različitih podzadataka je jedno od najvećih ograničenja za dobivanje dobrih performansi paralelnog programa.

Podzadaci u paralelnim programima često se nazivaju nitima. Neke paralelne računalne arhitekture koriste manje, lakše verzije niti, poznate kao vlakna, dok drugi koriste veće verzije poznate kao procesi. Ipak, „niti“ (engl. threads) se općenito prihvaćaju kao opći nazivi za podzadatke.

Paralelno programiranje na rješavanju problematike poreznih evidencija odnosi se na primjenu niti pri manipulaciji varijabli tijekom poreznih obračuna i izvještavanja. Ovo pruža ispravno izvršenje paralelnog programa. Ovaj vid okidača ovisit će o postavkama iz mrežnog prikaza (slika 5).

Najbitniji dio paralelnog programiranja je da jedna nit ima jedan ključ otključavanja ili jedan ključ zaključavanja, u protivnom uvodi se mogućnost da program završi u potpunom zastoju. (engl. deadlock).

Ovakvo postavljanje aplikacije za porezne evidencije u vidu paralelnog programiranja na način da podzadaci rade sinkrono, što zahtjeva korištenje barijera ili stop opcije. Ovakve stop opcije se uobičajeno implementiraju korištenjem softverskog zaključavanja. Postavljeni model paralelnog izvršavanja poreznih izvještaja putem poreznih varijabli u datumskoj opciji omogućuje slijedno izvođenje programa.

Sekvencijalna dosljednost je svojstvo paralelnog programa da njegovo paralelno izvođenje dovodi do istog rezultata kao i kod slijednih programa. Program je sekvencijalno dosljedan ako je zadovoljen slijedeći uvjet: „Rezultat ikogjeg izvršavanja je isti kao da su se operacije svih procesora izvele u sekvencijalnom slijedu, i da se operacija svakog pojedinačnog procesora izvela u slijedu određenim programom“ [12].

## 6. STANJE U RH I SUSJEDNIM ZEMLJAMA

Stanje u RH i susjednim zemljama je vrlo slično, zbog česte izmjene strukturiranih izvještaja, čestih izmjena poreznih stopa, ovaj dio softverskog inženjerstva koji je iz domene poslovnih aplikacija mogli bi se svrstati u sferu softverskog održavanja.

Zbog vrlo malog i uskog tržišta, poslovni subjekti koji izrađuju poslovne aplikacije pribjegavaju korektivno-adaptivnoj metodi održavanja aplikacija, kako bi se zadovoljile Zakonske norme i pravila, kao pretpostavka za produljenje životnog vijeka validnosti aplikacije uzrokovanih Zakonskim izmjenama.

Razloge ovakvom pristupu možemo naći u ekonomskim efektima naplate ovakvih usluga od strane autora poslovnih aplikacija, dok je ne posjedovanje ispravnih poreznih evidencija podložno visokim novčanim kaznama, te su korisnici izravno povezani sa autorima poslovnih aplikacija.

Temeljem ove činjenica, produljenje validnosti poslovnih aplikacija za praćenje porezno-poslovnih aplikacija u praksi se vrši na slijedeće načine:

- ustupanje novih modula aplikacija za praćenje poreznih evidencija korisnicima putem web sučelja,
- ustupanje novih modula za praćenje poreznih evidencija aplikacija putem elektroničke pošte,
- izravni dolazak djelatnika tvrtki autora ili distributera informacijskih aplikacija na lokaciju implementirane aplikacije.

Motivacije za uvođenje promjena su:

- unaprijediti kvalitetu knjigovodstvenih aplikacija,
- potaknuti razvoj knjigovodstvenih aplikacija sa standardnim modulom koji će rješavati probleme promjena poreznih Zakona,

- povećati konkurentnost aplikacija koje će u svom sastavu posjedovati module za slučaj promjena poreznih Zakona.

Glavni pokretači za uvođenje promjena su:

- povećanje brzine reakcije na vrijeme ažuriranja promjena, koji su uz primjenu novih modela trenutni,
- izbjegavanje mogućih pogrešaka na aplikaciji,
- povećanje autonomije rada korisnika na aplikaciji,
- povećanje zadovoljstva i motivacije korisnika.

## 7. ZAKLJUČAK

Kao rezultat provedenog istraživanja, uočeni su dugoročni trendovi u poreznim politikama koje se međusobno prožimaju i imaju uzročno posljedične veze sa trendovima u IT sektoru. Ti trendovi dovode do sve veće potrebe za raznovrsnim aplikacijama za praćenje poreznih evidencija koje će moći na odgovarajući način pohraniti, obraditi i prezentirati generirane podatke korisnicima na različitim računalnim platformama, a sve u skladu za odgovarajućim i važećim poreznim propisima.

Slijedni algoritam kao način razvoja poreznih aplikacija programiranjem, teško može udovoljiti ovim potrebama zbog malog broja stručnog programerskog kadra, čestih promjena poreznih stopa, dugog razvojnog ciklusa i složenog procesa razvoja i održavanja koji u uvjetima izmjene poreznih stopa imaju primarnu ulogu.

Kao rješenje ovakvog problema nameće se razvoj poreznih aplikacija u paralelnom modelu, koji ima zadaću i potencijal značajno pojednostaviti i skratiti održavanje, produljiti životni vijek poreznih aplikacija, značajno skratiti vrijeme implementacije i povećati i produljiti validnost poreznih aplikacija.

Nakon provedene analize trendova u IT i poreznim politikama na ovom području, može se zaključiti kako bi trend razvoja poreznih aplikacija u paralelnom modelu morao biti sve izraženiji. Aplikacije za praćenje poreznih propisa poželjno je da budu izrađene u paralelnom modelu i poželjno bi bilo da su multiplatforme. Ona programska rješenja i alati koji na jednostavan način budu ponudili veći broj navedenih kriterija za praćenje poreznih propisa, imaju veliku šansu da postanu standard na ovom području poslovnih softvera.

Očekuje se u bliskoj budućnosti uzlet novih poreznih aplikacija u paralelnom modelu ili značajne nadogradnje postojećih alata, koji će postati opće prihvaćeni od velike baze korisnika.

## REFERENCE

- [1] G. von Laszewski, "An interactive parallel programming environment applied in atmospheric science", Workshop Making its Mark - The Use of Parallel Processors in Meteorology, Reading, 1996.
- [2] P. Kacsuk, J. C. Cunha, G. Dozsa, J. Lourenco, T. Fadgyas, T. Antao, "A graphical development and debugging environment for

- parallel programs", *Parallel Computing - Special issue: distributed and parallel systems: environments and tools*, vol. 22, no. 13, 1997, pp. 1747-1770.
- [3] D. Banerjee, J. C. Browne, "Complete parallelization of computations: integration of data partitioning and functional parallelism for dynamic data structures", *Proceedings of IPPS '96, The 10th International Parallel Processing Symposium*, 1996, pp. 354-360.
  - [4] David B. Kirk, Wen-mei W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series)*, 2nd Edition, Morgan Kaufmann, 2012.
  - [5] J. C. Browne, S. I. Hyder, J. Dongarra, K. Moore, P. Newton, "Visual programming and debugging for parallel computing", *IEEE Parallel & Distributed Technology: Systems & Technology*, vol. 3, no. 1, 1995, pp.75-83.
  - [6] W. M. Johnston, J. R. P. Hanna, R. J. Millar, "Advances in dataflow programming languages", *ACM Computing Surveys (CSUR)*, vol. 36, no. 1, 2004, pp. 1-34.
  - [7] Eisenecker, U.: "Generative Programming (GP) with C++", *Proceedings of Modular Programming Languages (JMLC'97, Linz, Austria, March 1997.)*, Springer-Verlag, heidelberg 1997."
  - [8] Institut za javne financije, Smičiklasova 21, 10000 Zagreb, Aktuelni osvrt br.51 od 24.03.2013.godine
  - [9] David A. Patterson i John L. Hennessy. *Computer Organization and Design (Drugo izdanje)* Morgan Kaufmann Publishers, 1998. ISBN 1558604286, p. 715.
  - [10] Ratko Zelenika: „Metodologija i tehnologija izrade znanstvenog i stručnog djela“ Ekonomski fakultet ; Kastav : IQ plus, 2011
  - [11] Ivanović: „Metodologija znanstvenog istraživanja“, Kastav, SAIVA, 2011
  - [12] Reevaluating Amdahl's Law *Communications of the ACM* 31(5), 1988, pp. 532–33.
  - [13] Kovačić, Bosilj Vukšić, HERNANDEZ: „Upravljanje poslovnim procesima: organizacijsko i informacijski pristup“, Zagreb, Školska knjiga, 2008
  - [14] Sačer-Mamić, Žager: „Računovodstveni informacijski sustavi“, Hrvatska zajednica računovođa i financijskih djelatnika, Zagreb, 2007.(odabrana poglavlja),
  - [15] Javorović, Bilandžić: „Poslovneinformacije businessintelligence“, Goldenmarketing Tehnička knjiga, Zagreb, 2007. (odabranapoglavlja)
  - [16] Pavlić Mile: „Informacijski sustavi“, Zagreb, Školska knjiga, 2011
  - [17] Kaplan B., Harris-Salamone, K., Health IT success and failure: Recommendations from literature and AMIA workshop, *Journal of the American Medical Informatics Association* 16(3), 2009.: 291-299.
  - [18] Gheorghiu F., Why companies fail on the way to implementing project management methodology, *Project Management Today* 8(10),2006.
  - [19] Pavlić, M. (1990). Sistem analiza i modeliranje podataka. *Naučna knjiga, Beograd*.
  - [20] G.S. Almasi and A. Gottlieb. *Highly Parallel Computing*. Benjamin-Cummings publishers, Redwood city, CA, 1989.
  - [21] BARABÁSI, Albert-László; ALBERT, Réka. Emergence of scaling in random networks. *science*, 1999, 286.5439: 509-512.
  - [22] Gomez-Cabrero, D., Menche, J., Cano, I., Abugessaisa, I., Huertas-Migueláñez, M., Tenyi, A., ... & Tegnér, J. (2014). Systems Medicine: from molecular features and models to the clinic in COPD. *Journal of translational medicine*, 12(Suppl 2).
  - [23] ALAM, Maksudul; KHAN, Maleq. Parallel Algorithm for Generating Massive Random Networks with a Given Sequence of Expected Degrees. *arXiv preprint arXiv:1406.1215*, 2014.
  - [24] DOBRE, Ciprian; XHAFI, Fatos. Parallel programming paradigms and frameworks in big data era. *International Journal of Parallel Programming*, 2014, 42.5: 710-738.
  - [25] RADOŠEVIĆ, Danijel; MAGDALENIĆ, Ivan. Source code generator based on dynamic frames. *Journal of Information and Organizational Sciences*, 2011, 35.1: 73-91.
  - [26] SMITH, Adam M.; MATEAS, Michael. Answer set programming for procedural content generation: A design space approach. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2011, 3.3: 187-200.
  - [27] THE BARABÁSI-ALBERT MODEL 42 A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509-512, 1999.
  - [28] G.S. Almasi and A. Gottlieb. *Highly Parallel Computing*. Benjamin-Cummings publishers, Redwood city, CA, 1989.