

Optimizacija digitalne grafičke pripreme korištenjem skriptnih jezika

Brekalo, Sanja

Doctoral thesis / Doktorski rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:510068>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-19**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)





Sveučilište u Zagrebu

Grafički fakultet

Sanja Brekalo

**OPTIMIZACIJA DIGITALNE GRAFIČKE
PRIPREME KORIŠTENJEM
SKRIPTNIH JEZIKA**

DOKTORSKI RAD

Mentor:
Dr. sc. Klaudio Pap

Zagreb, 2015.



University of Zagreb

Faculty of Graphic Arts

Sanja Brekalo

DIGITAL PREPRESS OPTIMISATION USING SCRIPTING LANGUAGES

DOCTORAL THESIS

Supervisor:
Dr. sc. Klaudio Pap

Zagreb, 2015

UDK BROJ: 655.2:004.91

Povjerenstvo za ocjenu doktorske disertacije:

1. emeritus prof. dr. sc. Vilko Žiljak, Sveučilište u Zagrebu Grafički fakultet, predsjednik,
2. prof. dr. sc. Antun Koren, Sveučilište u Zagrebu Grafički fakultet, član,
3. izv. prof. dr. sc. Mario Barišić, Sveučilište u Osijeku Filozofski fakultet, vanjski član.

Povjerenstvo za obranu doktorske disertacije:

1. emeritus prof. dr. sc. Vilko Žiljak, Sveučilište u Zagrebu Grafički fakultet, predsjednik,
2. prof. dr. sc. Antun Koren, Sveučilište u Zagrebu Grafički fakultet, član,
3. izv. prof. dr. sc. Mario Barišić, Sveučilište u Osijeku Filozofski fakultet, vanjski član,
4. prof. dr. sc. Nikola Mrvac, Sveučilište u Zagrebu Grafički fakultet, zamjenski član,
5. doc. dr. sc. Zvonimir Sabati, Sveučilište u Zagrebu Fakultet informacijskih znanosti u Varaždinu, zamjenski vanjski član.

Datum obrane doktorske disertacije: 27. studenoga 2015.

Mjesto obrane doktorske disertacije: Sveučilište u Zagrebu, Grafički fakultet

Povjerenstvo za obranu doktorske disertacije donijelo je sljedeću odluku:

„Obrabila– jednoglasnom odlukom Povjerenstva“

INFORMACIJE O MENTORU: red. prof. dr. sc. Klaudio Pap

Dr. sc. Klaudio Pap redovni je profesor na Grafičkom fakultetu Sveučilišta u Zagrebu. Rođen je 20. ožujka 1963. godine u Zagrebu. Nakon završene matematičke gimnazije studirao je na Elektrotehničkom fakultetu Sveučilišta u Zagrebu gdje je i diplomirao na smjeru Računarske tehnike 1988. godine. Magistrirao je 1997. godine na istom fakultetu (sada Fakultet elektrotehnike i računarstva) na smjeru Računarskih znanosti s temom "Simulacija protoka informacija između računala i digitalnog tiskarskog stroja" (mentor dr. sc. Gabro Smiljanić). Doktorirao je 25. veljače 2004. na Fakultetu elektrotehnike i računarstva na smjeru Računarskih znanosti s temom "Simulacija hibridnih i digitalnih sustava sa sučeljima za obradu slikovnih elemenata i rastera" (mentor dr. Vilko Žiljak).

U zvanje znanstvenog suradnika izabran je na zagrebačkom sveučilištu 1. prosinca 2004. u području tehničkih znanosti, polju grafička tehnologija te postaje docent za kolegije Računarski slog i Računarske grafike. Na istom sveučilištu je 18. siječnja 2010. izabran u znanstveno zvanje viši znanstveni suradnik, a 12. travnja 2010. postaje izvanredni profesor u području tehničkih znanosti, polju grafička tehnologija. Znanstveni savjetnik na Sveučilištu u Zagrebu postaje 18. listopada 2011. u području tehničkih znanosti, polju grafička tehnologija. Dana 26. veljače 2005. godine postaje član suradnik Akademije tehničkih znanosti Hrvatske. Dužnost dekana Grafičkog fakulteta Sveučilišta u Zagrebu obavlja od 1. siječnja 2015. do danas.

Tijekom svog rada bavi se istraživanjem, razvojem i primjenom računala. Područja na kojima intenzivno radi su računarska grafika, procesiranje slike i teksta, modeliranje i simuliranje s računalom, web tehnologije, digitalni tisak i grafički programski jezici. Sudjelovao je na uvođenju novih tehnologija o čemu je održao niz seminara, predavanja i tečajeva. Koautor je pet (5) razvojnih proizvoda, pet (5) softverskih paketa, a primio je tri (3) znanstvene nagrade. Objavio je šest (6) znanstvenih članaka u časopisima citiranim u tercijarnim publikacijama i osamnaest (18) znanstvenih radova u časopisima citiranim u sekundarnim publikacijama. Autor je pet (5) knjiga kao i mnogih radova na međunarodnim znanstvenim skupovima. Godine 2011. Hrvatski ga je sabor proglasio dobitnikom Državne nagrade za znanost za 2010. u području tehničkih znanosti za znanstveno otkriće „Kreiranje tiskarskih boja za vidljivi i infracrveni spektar“, a Hrvatska akademija tehničkih znanosti dobitnikom godišnje nagrade "Rikard Podhorsky" za 2010.

ZAHVALA

Posebno zahvaljujem svome mentoru izv. prof. dr. sc. Klaudiju Papu na savjetima, susretljivosti, uloženom trudu i nadasve pomoći oko izrade ove disertacije.

Zahvalila bih i grafičkim inženjerima te djelatnicima grafičke pripreme koji su sudjelovali pri izradi unaprjeđenja, na njihovim prijedlozima i uloženom vremenu.

Na kraju bih zahvalila svojoj obitelji, nadasve majci Katarini jer mi je bila najveća podrška u svemu što radim, na njezinoj dugogodišnjoj bezrezervnoj ljubavi, razumijevanju i prijateljstvu.

SAŽETAK

Implementiranjem novih mogućnosti u postojeće programe za stolno izdavaštvo moguće je optimizirati procese digitalne grafičke pripreme. Skriptni programi koji izvršavaju izradu prijeloma, pod određenim uvjetima, mogu ubrzati i automatizirati digitalnu grafičku pripremu te smanjiti broj pogrešaka kod formatiranja dokumenta. Temeljem analize proizvodnih procesa izneseni su kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju korištenjem skriptnih tehnologija. Predloženi su novi modeli za rješavanje složenih zadataka grafičke pripreme kojima se unaprjeđuju postupci ubacivanja, zamjene i oblikovanja teksta i slika, ispravljanja pogrešaka u tekstu, izrade automatiziranog prijeloma periodičkih publikacija, automatizacije radnih procesa baziranih na XML-u, izrade montaže u programu za prijelom, individualizacije i varijabilne grafičke pripreme. U radu se novi izrađeni modeli za poboljšanje procesa u grafičkoj pripremi vrednuju naspram standardnih procesa te se određuje nivo optimizacije mjerenjem vremena, određivanjem stupnja postignute automatizacije te određivanjem pouzdanosti standardnih i unaprjeđenih modela rada. Iz izvedenog istraživanja donijeti su zaključci o uštedama i prednostima koje se dobivaju upotrebom skriptnih tehnologija te uvjeti za optimalnu upotrebu u standardnim programima i metodama rada.

Ključne riječi: digitalna grafička priprema, skriptiranje, ExtendScript, optimizacija

SUMMARY

For quite some time the printing business has been under heavy pressure because of overcapacity, dropping prices and the delocalization of the production to low income countries. The graphic communications industry, like most other "manufacturing" industries has increased pressure to drive cost down while simultaneously providing automation. Every step in the graphic communications value chain is under scrutiny and a candidate for refinement. To survive in this competitive world, printers have to invest in tools that reduce production costs, create additional value for their customers and automate production processes. Automation of workflow processes can provide tremendous value both in terms of operational efficiency and revenue growth. The benefit of automation in graphic prepress lies primarily in an increase of efficiency in the production of layout and imposition of repetitive tasks by shortening lead time. Another benefit is cost reduction, since more work can be done with fewer staff members.

Prepress processes can be optimised by implementing new features in the existing desktop publishing programs. Scripting programs that execute layout process under certain conditions can accelerate, automate and reduce the number of errors in digital prepress when formatting documents. The main purpose of these theses is to spread the idea about computer work optimisation in graphic prepress workflows. This research proves that improvements in graphic prepress processes can be obtained by implementing new suggested models which use scripting languages. Scripts can be written to execute just one task, complex scripts can perform multiple tasks and some scripts automate the entire publishing process.

New models for solving complex tasks are proposed to enhance procedures of inserting, replacing and formatting text and images, correcting errors in the text, making automated layout of periodicals, automation of work processes based on XML, making impositions in the layout program, individualization and variable prepress. Quantitative research was carried out to determine achieved automation and time reduction by using optimised processes. Prepress models for process improvement are evaluated against standard processes, and the level of optimization is determined by measuring time, automation and reliability levels of standard and enhanced models of operation. This proves the hypothesis that features of workflow processes which quantify savings and benefits can be determined.

From conducted analysis of digital prepress 10 new script workflow models were made. Each model was evaluated against standard production of the task. By graphic prepress

analysis and script execution evaluation findings about criteria in process selection for optimisation with scripting technologies were made. Thesis also highlights savings and benefits of using scripting technologies in graphic prepress. Thesis proves that by replacing only certain tasks it is possible to automate digital prepress processes since there are tasks that cannot be optimised. By analysis such tasks were described.

Special interest is given to the hypothesis which states that by adding new script models to graphic prepress complex tasks that are otherwise hard to address can be optimally achieved. The hypothesis was proven by implementing scripts that add new functionality to the production in standard prepress programs.

With scripts that execute complex tasks execution period can be time consuming. This paper also deals with written script optimization in order to shorten the time of its execution and to identify factors that affect the script execution time.

Key words: digital graphic prepress, scripting, ExtendScript, optimisation

SADRŽAJ

SAŽETAK	V
SUMMARY	VI
SADRŽAJ	VIII
1. UVOD	1
2. CILJ I PRETPOSTAVKE ISTRAŽIVANJA	6
3. TEORIJSKE PRETPOSTAVKE	7
3.1. Skriptiranje u DTP programima	7
3.1.1. Adobe InDesign skriptiranje	8
3.1.2. Adobe InDesign, JavaScript i objektni model.....	9
3.2. XML i SVG	12
4. EKSPERIMENTALNI DIO	13
4.1. Metode rada i eksperimentalni okvir	13
4.2. Analiza problema i analiza izvedivosti pojedinog modela u okviru grafičke pripreme.....	14
4.2.1. Kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju	14
4.2.2. Analiza procesa digitalne grafičke pripreme.....	16
4.3. Izrada prijedloga modela za izvođenje unaprjeđenja uz pomoć skriptnih jezika i njihovo implementiranje u definirani proces grafičke pripreme	18
4.3.1. Skripta za brisanje neželjenih bjelina u tekstu (oznaka: S1).....	20
4.3.2. Skripta za zadržavanje uvezenih znakovnih stilova (oznaka: S2).....	22
4.3.3. Skripta za uređivanje tablica u periodičkoj publikaciji (oznaka: S3).....	24
4.3.4. Skripta za izradu numeracije i montaže kartica (oznaka: S4)	29
4.3.5. Skripta za izradu montaže i personalizacije kartica iz podataka u XML datoteci (oznaka: S5).....	35
4.3.6. Skripta za izradu montaže u programu za prijelom (oznaka: S6)	40
4.3.7. Skripta za automatizaciju izrade ePub dokumenta koji sadrži SVG datoteke (oznaka: S7).....	44
4.3.8. Skripta za automatsku obradu fotografija (oznaka: S8)	49
4.3.9. Skripta za izradu barkoda iz varijabilnih podataka (oznaka: S9).....	54
4.3.10. Skripta za obilježavanje ispravaka u tekstu nakon lekture (oznaka: S10)	59

4.4.	Mjerenje optimizacije i obrada rezultata	63
4.4.1.	Mjerenje vremena izvršavanja zadataka grafičke pripreme standardnim i skriptnim načinom.....	63
4.4.2.	Mjerenje automatizacije izvršavanja zadataka grafičke pripreme skriptnim izvođenjem	63
4.4.3.	Mjerenje pouzdanosti izvršavanja zadataka grafičke pripreme skriptnim izvođenjem u usporedbi sa standardnim izvođenjem.....	64
5.	REZULTATI	67
5.1.	Rezultati mjerenja skripte za brisanje neželjenih bjelina u tekstu (S1).....	67
5.2.	Rezultati mjerenja skripte za zadržavanje uvezenih znakovnih stilova (S2).....	71
5.3.	Rezultati mjerenja skripte za uređivanje tablica u periodičkoj publikaciji (S3).....	75
5.4.	Rezultati mjerenja skripte za izradu numeracije i montaže kartica (S4)	78
5.4.1.	Optimizacija skripte za kraće vrijeme izvođenja	78
5.4.1.1.	Početna vremena izvođenja skripte	79
5.4.1.2.	Testiranje <i>doScript</i> metode postavljanjem <i>UndoModes</i> parametra funkcije.....	79
5.4.1.3.	Korištenje povezanih (<i>eng. linked</i>) umjesto samostalnih tekstualnih okvira	81
5.4.1.4.	Funkcija „ <i>pad</i> “	85
5.4.1.5.	Korištenje <i>everyItem()</i> za dohvaćanje podataka u objektima kolekcije (<i>eng. Collection objects</i>).....	89
5.4.1.6.	Testiranje brzine izvršavanja <i>for</i> i <i>while</i> petlji.....	90
5.4.1.7.	Konačna optimizacija skripte	91
5.4.2.	Rezultati usporedbe standardnog i skriptnog načina izvođenja zadatka	95
5.5.	Rezultati mjerenja skripte za izradu montaže i personalizacije kartica iz podataka u XML datoteci (S5).....	99
5.6.	Rezultati mjerenja skripte za izradu montaže u programu za prijelom (S6)	105
5.7.	Rezultati mjerenja skripte za automatizaciju izrade ePub dokumenta koji sadrži SVG datoteke (S7).....	110
5.8.	Rezultati mjerenja skripte za automatsku obradu fotografija (S8)	114
5.9.	Rezultati skripte za izradu barkoda iz varijabilnih podataka (S9).....	118
5.10.	Rezultati skripte za obilježavanje ispravaka u tekstu nakon lekture (S10)	121

6. DISKUSIJA	122
6.1. Optimizacija vremena izvršavanja zadataka grafičke pripreme	122
6.2. Automatizacija izvršavanja zadataka grafičke pripreme	128
6.3. Pouzdanost optimalnog i ispravnog izvršenja zadatka	131
6.4. Optimalno rješavanje zahtjevnih zadataka koji su teški za ostvarivanje.....	135
6.5. Kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju korištenjem skriptnih tehnologija	137
6.6. Uštede i prednosti koje se dobivaju korištenjem skriptnih tehnologija.....	140
6.7. Optimalno korištenje skriptnih tehnologija u standardnim programima i metodama rada.....	142
6.8. Zadatci grafičke pripreme za koje nije optimalno korištenje skriptnih tehnologija	144
7. ZAKLJUČAK.....	146
LITERATURA	148
PRILOZI.....	154
ŽIVOTOPIS.....	164

1. UVOD

Grafička se tehnologija razvija u hibridnu tehnologiju koja kombinira konvencionalni i digitalni tisak te inženjerstvo informacijskih tehnologija. Tiskarska kuća je idealno mjesto za generiranje spona između različitih tehnologija jer već posjeduje visokokvalitetne resurse potrebne za izradu istih [1].

Iz povijesnog se razvoja grafičke pripreme primjećuje velik utjecaj tehnologija na ubrzanje i optimizaciju procesa. Promjene koje se događaju danas, nisu toliko dramatične kao one koje su se dogodile sredinom 1980-ih godina. Neke od promjena koje se i danas događaju s aspekta optimizacije proizvodnih procesa uključuju:

- razvoj računala koja konstantno postaju sve brža i jača omogućavajući izvođenje intenzivnijih procedura;
- razvoj digitalnih komercijalnih tiskarskih strojeva koji omogućuju cjenovno i vremenski prihvatljiv tisak manjih naklada, iako ponekad pod cijenu manje kvalitete;
- elektronsko odobrenje tiskanog materijala (u PDF formatu) što ubrzava cijeli proizvodni proces, osobito kod jednobojskih poslova; kod višebojskih poslova prednosti se dobivaju uporabom *Color Management* tehnika;
- konstantan razvoj DTP programa izlaskom novih inačica.

Grafička priprema i tiskarske usluge već su neko vrijeme pod velikim pritiskom zbog prekapacitiranja, pada cijena i seljenja poslova u zemlje s niskim primanjima. Kako bi grafička poduzeća bila konkurentna, tiskare moraju uložiti u alate koji smanjuju troškove proizvodnje te stvaraju dodanu vrijednost za njihove klijente [2].

Temeljni je cilj promjena koje se događaju u grafičkoj pripremi, daljnja optimizacija radnih procesa te dodavanje novih mogućnosti u postojeće radne procese. Imperativ postaje smanjenje troškova, uz smanjenje radne snage, vremena izrade i broja pogrešaka. U razvoju i nadogradnjama programa za grafičku pripremu također možemo pratiti djelomična ostvarenja navedenih ciljeva. Tako je 1985. godine Aldus, tvrtka koju je kasnije kupio Adobe, izdala prvi DTP software koji se zvao PageMaker. On je omogućio dizajnerima prijelom stranica u WYSIWYG¹ modu umjesto pisanja naredbi u fotoslogu. Ovaj alat je uskoro zamijenjen programom QuarkXpress tvrtke Quark. U današnje vrijeme standardom za grafičku pripremu postaje InDesign tvrtke Adobe koji je polako izazivao i naposljetku počeo preuzimati Quark-

¹WYSIWYG (engl. *what you see is what you get*) - što vidiš to ćeš i dobiti

ovu dominaciju u DTP industriji. Danas tržišna zastupljenost i upotreba nisu poznati jer InDesign često dolazi u Adobe Suite paketima zajedno s ostalim programima koji se upotrebljavaju za grafičku pripremu pa je teško odrediti njegovu stvarnu uporabu [3]. Navedeni programi donose poboljšanja u smislu ubrzavanja procesa, automatizacije i smanjenja pogrešaka ugrađenim alatima za automatizaciju zadataka izrade prijeloma.

Jedna od mogućnosti dodana u navedene programe je i skriptiranje. Skriptiranje u DTP programima nije novost. Već su ranije verzije današnjih standardnih DTP programa imale mogućnost izvršavanja skripti i omogućavale automatizaciju radnih procesa korištenjem skripti. Poznato je da se određeni zadatci grafičke pripreme mogu rješavati skriptnim tehnologijama, ali se ne zna kako te tehnologije optimalno koristiti ili ih možda hibridno povezivati sa standardnim programima i metodama rada.

Većina tvrtki koje se bave grafičkim uslugama posjeduje neke od standardnih programa koji se svakodnevno koriste u prijelomu publikacija, koji odgovaraju na navedene zahtjeve. Preduvjet za pisanje skripte je poznavanje jednog od podržanih skriptnih jezika i objektnog modela programa za koji se skripte pišu. To može biti problem za grafičkog dizajnera koji ne posjeduje osnovna znanja programiranja pa iz navedenog razloga ne primjenjuje skriptne metode rada. Problem kod implementacije skriptnih modela može biti i nedovoljno poznavanje samog programa i njegovih mogućnosti [4].

Skriptiranjem se mogu izrađivati kratke skripte koje odrađuju samo jedan zadatak, kompleksne skripte koje obavljaju višestruke zadatke, pa sve do onih koje automatiziraju gotovo cijeli izdavački proces [5].

Postoje zadatci koji se mogu ostvariti samo skriptiranjem u programu za prijelom, primjerice personalizacija izradom barkoda, numeracija ili automatska montaža dokumenta u programu za prijelom. Tako se upotrebom postojećih programskih alata može izraditi priprema i montaža dokumenata za tisak koji sadrže varijabilne podatke kao što je barkod čiji se sadržaj mijenja po kartici koja se montira. Varijabilna grafička priprema naziva se i tiskom na zahtjev te omogućuje personalizaciju grafičkih proizvoda. Postoje mnogi komercijalni programski alati koji omogućuju potrebne funkcije za pripremu proizvodnje varijabilnih digitalnih otisaka, ali mnogi su ili preskupi ili su ograničeni na specijalne vrste digitalnih strojeva. Za manje tvrtke koje se bave tiskom ili pakiranjem takav trošak može se izbjeći korištenjem skriptnih tehnologija i alata koje već posjeduju. Upravo su takvi zadatci zanimljivi za istraživanje u ovom radu.

Skriptiranje je opcija koja se može koristiti kako bi se implementirali novi modeli kod varijabilnog tiska na strojevima za digitalni tisak uz istovremenu automatizaciju procesa [6].

Tehnologija automatske montaže i prijeloma predstavlja veliku vrijednost konačnim korisnicima alata za digitalnu pripremu jer može ukloniti zastoje uzrokovane izradom dokumenata sačinjenih od vrlo prilagođenih tekstualnih i slikovnih sadržaja. Navedeno predstavlja vrlo zahtjevan tehnički problem zbog toga što uključuje dvodimenzionalnu optimizaciju pozicija i dimenzija višestrukih tipova sadržaja kao što su slike, tekst i vektorske grafike [7]. Na ovom području rađena su opsežna istraživanja. Jedno od ranijih nastojanja može se pripisati izradi *Juno-2* editora za prijelom kojega su razvili Heydon i Nelson u ranim devedesetim, a koji je bio baziran na tehnikama zadovoljenja ograničenja² [8][9]. Jacobs et. al. [10] uvode sustav za prijelom dokumenata koji je prilagodljiv; automatski bira najbolji predložak za dati sadržaj. Purvis et. al. [11] formaliziraju kreaciju personaliziranih dokumenata kao višekriterijsku optimizaciju³ [12] te koriste genetski algoritam⁴ koji vrši automatski prijelom dokumenata. Zadovoljavanje ograničenja, učenje i genetski algoritmi tehnike su koje su istraživane u izradi automatiziranih sustava prijeloma [13] [14]. Johari et. al. [15] izradili su specijalizirani sustav za paginaciju i prijelom telefonskog imenika. Berkner et. al. [16] uveli su metodu inteligentnog skaliranja slika i dijelova teksta pomoću informacija dostupnih u JPEG2000 datoteci. Izrađeni su i prijedlozi algoritama koji automatiziraju prijelom i dizajn slika u foto albumima [17] [18].

Mnogi istraživači u polju automatiziranog prijeloma smatraju problematičnom činjenicu da su mnogi prijelomi izrađeni ručno od strane dizajnera [11] [13] [19] [20] što je dovelo do mnogih istraživanja u automatizaciji prijeloma informacija. Najčešća tehnika za izradu automatskih prijeloma je upotreba predložaka (engl. *Templates*). Na tržištu su dostupna programska rješenja koja automatiziraju prelamanje dokumenta i izradu gotovih stranica prema pravilima predložaka koji se izrađuju ručno. Postoje i rješenja bazirana na skriptama i dodacima (engl. *Plug-ins*) za standardne DTP programe.

²Zadovoljenje ograničenja (engl. *Constraint satisfaction*) koristi se najčešće pri izradi sustava umjetne inteligencije u kojima se problemi promatraju kao skup ograničenja pri odlučivanju. Tehnikom zadovoljenja ograničenja pronalaze se rješenja na set ograničenja koja zadaju uvjete koje varijable moraju zadovoljiti. Rješenje je tako set vrijednosti za varijable koje zadovoljavaju sva ograničenja.

³Metode višekriterijske optimizacije (engl. *Multi-objective Optimization*) i odlučivanja koriste se u problemima odlučivanja uz postojanje više uobičajeno konfliktnih kriterija, tj. više funkcija cilja. Ne može se govoriti o najboljoj niti optimalnoj odluci u klasičnom smislu, već o odluci koja, na neki način, najbolje zadovoljava postavljene kriterije.

⁴Genetički algoritmi (engl. *Genetic algorithms*) skup su tehnika koje se temelje na postavkama izvedenim iz prirodne genetike populacije. Primjenjuju se u rješavanju problema u inženjerstvu, umjetnoj inteligenciji i ostalim znanstvenim poljima.

Neke od prednosti automatske publikacije i prijeloma su:

- povećanje proizvodnje smanjenjem vremena pripreme dokumenta [20];
- smanjenje troška proizvodnje jer se više poslova može obaviti s manje radne snage;
- potencijal za automatsku generaciju personaliziranih dokumenata uz mogućnost ispisa na digitalnim strojevima za tisak [19];
- automatizirani poslovi mogu se odvijati u neradnim vremenima poduzeća [21].

Uređivanje velike količine teksta i slika te prelamanje sadržaja primjeri su poslova koji bi se trebali automatizirati jer su takvi zadatci ponavljajući, kod njih lako može doći do pogrešaka i zamorni su kada se provode manualno. Neki do sad utvrđeni kriteriji za odabir poslova u automatskom prijelomu bili bi:

- dizajn je baziran na pravilima ili predlošcima, s jednostavnim ili ponavljajućim prijelomom;
- materijal koji se prelama sastoji se od mnogo stranica;
- materijal se često objavljuje s podacima, tj. sadržajem ili dijelovima sadržaja koji dolaze iz baze podataka ili je izrađen pomoću skripte;
- posao je dugotrajan i ponavljajući, prezamoran da bi se odradio ručno [21].

U ovom radu primijenjen je drugačiji pristup automatizaciji. Umjesto automatizacije grafičke pripreme u cijelosti, odlučeno je automatizirati samo određene dijelove procesa prijeloma identificiranjem specifičnih poslova podobnih za automatizaciju. Takvim pristupom moguće je automatizirati dio procesa na individualnoj razini kako bi se povećala produktivnost.

Pretpostavljeno je da se analizom grafičke pripreme mogu odrediti procesi podobni za automatizaciju skriptiranjem i da ih je moguće zadovoljavajuće optimizirati. Cilj istraživanja je dokazati da je novim predloženim modelima koji koriste skriptne jezike, moguće grafičku pripremu pod određenim uvjetima optimizirati povećanjem brzine, skraćivanjem vremena rada, automatizacijom procesa i smanjenjem pogrešaka u formatiranju dokumenta.

Ovaj rad bavi se optimizacijom procesa grafičke pripreme upotrebom standardnih DTP programa, skriptiranja i XML baziranih tehnologija te prednostima i nedostacima ovakvih modela rada. Rezultati ovog znanstvenog rada daju znanja koja u praksi mogu povećati primjenu i način skriptnih tehnologija te se daju smjernice za njihov daljnji razvoj u programima grafičke pripreme. Izradom i testiranjem novih modela rada dokazano je da se

usprkos problemima implementacije mogu postići veća unaprjeđenja u procesima grafičke pripreme.

Tijekom istraživanja prepoznate su prednosti koje donosi upotreba skriptnih modela. Određeni su i kriteriji za odabir procesa koji želimo unaprijediti skriptiranjem. Automatizacija nije uvijek isplativa jer zahtijeva izradu skripti što je često dugotrajan proces [20]. Zbog toga je potrebno odluku o odabiru procesa za automatizaciju bazirati na analizi vremena potrošenog u grafičkoj pripremi. Rad ciljano iziskuje analizu problema u stvarnom kontekstu kako bi se dobile preporuke praktične vrijednosti. Unaprijed određeni kriteriji za odabir procesa bili su sljedeći: posao je ponavljajući i nije kreativan, a napisana skripta može se više puta koristiti za druge slične poslove. Kako bi se odabrali poslovi koje je potrebno automatizirati, a udovoljavaju navedenim kriterijima, izrađena je analiza proizvodnih procesa stvarnog poduzeća te izrada automatizacije prema utvrđenim potrebama. Takvim pristupom dobiva se optimizacija stvarnog mjerljivog procesa prilagođena potrebama proizvodnje. Zbog ovakvog je pristupa posao izrade skripti za automatizaciju jednostavniji. Naime, skripte ne moraju biti pretjerano općenite, nije potrebno zadovoljiti sve moguće scenarije, već samo one koji su dovoljni da zadovolje potrebe odabrane skupine korisnika. Time se skraćuje vrijeme izrade skripte.

Skripte koje zamjenjuju dugotrajne zadatke grafičke pripreme često i same traju dulji vremenski period. Kako bi se utvrdio utjecaj načina pisanja skriptnog programa, provedena su testiranja koja utvrđuju faktore koji utječu na skriptna vremena. Rezultati ovog rada daju smjernice za optimizaciju napisanog koda. Pri izradi rada nije bilo poznato kako se skriptne tehnologije optimalno koriste i mogu li se postići veće uštede u vremenu izvršavanja skripte njihovim optimiziranjem. Za skripte koje se kratko izvršavaju optimizacija gotovo da i nije potrebna, ali se, kako je utvrđeno istraživanjem, preporučuje za skripte čije izvršavanje traje dulji vremenski period. Optimizacijom samih skripti dobivamo vrijeme izvršavanja koje može biti drastično smanjeno. Pritom se smanjuje i vrijeme čekanja samog operatera da skripta izvrši potrebne zadatke kako bi mogao nastaviti s ostalim poslovima. U vrijeme izvršavanja skripte program u kojemu se skripta izvršava nije dostupan za ostale zadatke.

Rezultati ovoga rada dokazuju kako navedene tehnologije mogu doprinijeti ubrzanju i automatizaciji radnih procesa čineći ih pouzdanijim. U ovom radu istraženi su modeli rada te implementacija i mogućnosti koje donosi integracija tehnologija.

2. CILJ I PRETPOSTAVKE ISTRAŽIVANJA

Cilj istraživanja bio je dokazati da se novim predloženim modelima, koji koriste skriptne jezike, postižu znatna unaprjeđenja u procesima grafičke pripreme povećanjem brzine i pouzdanosti. U tu svrhu postavljene su sljedeće hipoteze istraživanja:

1. Temeljem analize segmenata proizvodnih procesa grafičke pripreme moguće je grafičku pripremu, upotrebom skriptnih jezika pod određenim uvjetima, optimalno ubrzati s povećanjem pouzdanosti i isplativosti;
2. Moguće je dodavanjem novih skriptnih modela u standardne programe grafičke pripreme optimalno rješavati zahtjevne zadatke koji su inače teški za ostvarivanje;
3. Zamjenom samo određenih radnih procesa sa skriptnim datotekama moguće je automatizirati procese grafičke pripreme jer postoje zadatci grafičke pripreme za koje nije optimalno korištenje skriptnih tehnologija;
4. Moguće je odrediti značajke radnih procesa pomoću kojih će se kvantificirati uštede i prednosti koje se dobivaju korištenjem skriptnih tehnologija.

Izvorni znanstveni doprinos disertacije očekuje se u sljedećem:

- Analiza segmenata proizvodnih procesa grafičke pripreme koji bi se mogli upotrebom skriptnih jezika pod određenim uvjetima optimalno ubrzati s povećanjem pouzdanosti i isplativosti;
- Novi skriptni modeli, samostalni i integrirani sa standardnim programima, za optimalno rješavanje zahtjevnih zadataka grafičke pripreme koji su inače teški za ostvarivanje;
- Definiranje kriterija za optimalnu upotrebu skriptnih tehnologija u standardnim programima i metodama rada;
- Vrednovanje učinkovitosti novih modela pomoću kvantificiranja postignutih ubrzanja, automatizacije i povećanja pouzdanosti radnih procesa u grafičkoj pripremi.

3. TEORIJSKE PRETPOSTAVKE

3.1. Skriptiranje u DTP programima

Skriptiranje je jedna od najmoćnijih mogućnosti u DTP programima namijenjenim za pripremu za tisak i e-publikacije. Nijednom drugom mogućnošću dostupnih alata ne može se toliko uštedjeti vrijeme, rad i novac. Skriptiranje je odličan alat za izvođenje različitih zadataka. Skripte su nizovi izjava koje upućuju aplikaciju kako da odradi set zadataka. One mogu biti jednostavne i automatizirati česte postupke ili kompleksne i davati nove mogućnosti u programu. Skripte možemo kreirati sami ili možemo izvoditi skripte koje su kreirali drugi [5]. Kako bismo iskoristili prednosti skriptiranja, potrebno je odabrati DTP program koji podržava pisanje i izvršavanje skripti. Na tržištu je ponuđeno više opcija pri odabiru alata za DTP. Gotovo svi ponuđeni alati omogućuju neki oblik skriptiranja i automatizacije proizvodnih procesa. Trenutno ponuđeni alati su Adobe InDesign i FrameMaker, tj. Adobe Creative Suite paketi s više alata, QuarkXPress, Serif PagePlus, Microsoft Office Publisher i Scribus [22].

Gotovo svi alati koji se upotrebljavaju za izradu grafičke pripreme iz Adobe Creative Suite paketa podržavaju pisanje skripti. Moguće je izrađivati skripte za Adobe Illustrator, Photoshop, InDesign, InCopy, Bridge te FrameMaker [23]. Jezik koji se koristi pri skriptiranju ovisi o skriptnom sustavu operativnog sustava na kojemu se skripta izvršava. AppleScript se koristi za Mac OS, VBScript za Windows, a JavaScript za bilo koji od navedenih operativnih sustava. Iako se ti skriptni jezici razlikuju, načini na koji rade su veoma slični pa je prevođenje jedne skripte u drugu vrlo jednostavno [24].

QuarkXPress podržava skriptiranje u AppleScript jeziku [25]. U usporedbi skriptiranja InDesigna i QuarkXPressa može se istaknuti da InDesign nudi prednosti u skriptiranju zbog podržavanja više skriptnih jezika i mogućnosti povezivanja sa sučeljem Adobe Photoshopa i Illustratora. InDesign daje nešto bolje rješenje s većom podrškom za korisnike i više dostupnih informacija na Internetu.

Serif PagePlus i Microsoft Office Publisher su po mogućnostima nešto slabiji alati [22] za izradu grafičke pripreme u usporedbi s programima Adobe InDesign i QuarkXPress. Serif PagePlus je dobro rješenje za osnovne zadatke dizajna i prijeloma stranica, ali još uvijek ne podržava skriptiranje. Microsoft Office Publisher podržava skriptiranje u obliku „*Macro*“ naredaba [26], kao i ostali alati iz Office paketa. „*Macro*“ naredbe su set instrukcija pisanih u Microsoft Visual Basic programskom jeziku.

Scribus je besplatan alat, mogućnostima vrlo sličan profesionalnim komercijalnim alatima za izradu prijeloma. „The Scripter“ je programski dodatak za Scribus koji omogućuje pisanje i izvršavanje automatizacijskih skripti. Skripte se pišu u Python programskom jeziku dostupnom za sve operacijske sustave [27].

3.1.1. Adobe InDesign skriptiranje

Adobe InDesign kao DTP program omogućuje izradu stranica za tisak i e-izdanja. Danas je jedan od standardnih i najpopularnijih alata za profesionalnu upotrebu u grafičkoj pripremi. Između brojnih mogućnosti koje nudi, dostupno je i izvođenje skripti koje omogućuju:

- formatiranje i prijelom teksta,
- unošenje, pretraživanje i izmjenu teksta,
- crtanje objekata na stranicama,
- ispisivanje ili izvoženje dokumenta u neki od formata grafičke pripreme,
- ubacivanje slika,
- izradu novih izbornika ili novih mogućnosti u postojećim izbornicima,
- kreiranje i prikazivanje dijaloških okvira i panela,
- izvođenje skripte ovisno o korisnikovoj selekciji,
- čitanje i zapisivanje u tekstualne datoteke,
- obrađivanje XML podataka,
- razmjenjivanje podataka s drugim aplikacijama.

U načelu, sve što se u programu može napraviti s korisničkim sučeljem, može se napraviti i sa skriptom. Postoje mogućnosti koje se mogu napraviti samo skriptiranjem, ali postoje i neke postavke programa koje nije moguće mijenjati preko skripti, a obično se tiču korisničkog sučelja, primjerice:

- postavljanje radnog okruženja (*eng. workspace*),
- definiranje prečaca tipki (*eng. shortcuts*),
- dodavanje novih tipova objekata u InDesign dokument,
- dodavanje novih osnovnih mogućnosti u program, kao što je novi algoritam, za kompoziciju teksta (*eng. text-composition engine*) i sl.

Za navedene mogućnosti potrebno je koristiti InDesign Software Development Kit (SDK) koji omogućuje pisanje kompleksnih dodataka (*eng. plug-ins*) uporabom C++ programskog jezika [24].

3.1.2. Adobe InDesign, JavaScript i objektni model

JavaScript je skriptni jezik najčešće korišten na internetu. Podržavaju ga svi poznatiji preglednici i stoga je popularan kod izrade web sadržaja. Cilj kreiranja JavaScript jezika bio je dodati interaktivnost HTML stranicama.

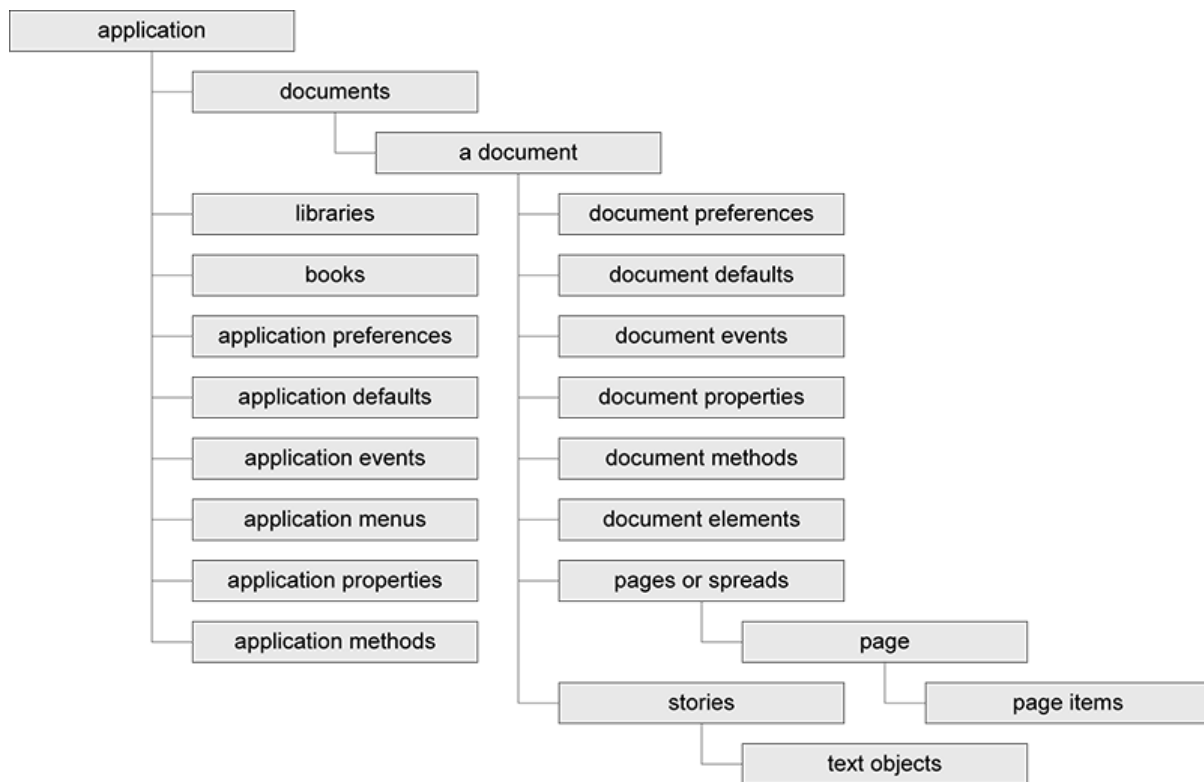
Skriptni jezici su programski jezici koji se sastoje od izvršnog računalnog koda. JavaScript je *interpreter* što znači da se skripta izvršava odmah, naredbu po naredbu, bez prethodnog prevođenja (kompiliranja, *eng. compile*) cijelog programa i kreiranja izvršne datoteke. On je javno raspoloživ skriptni jezik te nije potrebna licenca za njegovu upotrebu [28]. JavaScript se kao skriptni jezik brzo uči, u usporedbi s tradicionalnim programskim jezicima, te je često potrebno manje koda kako bi se izvršio određeni zadatak.

InDesign JavaScript podrška bazirana je na Adobe implementaciji JavaScripta poznatoj kao ExtendScript. Interpreter za ExtendScript odgovara trenutnim ECMA 262 standardima te su podržane sve jezične značajke JavaScript verzije 1.5 [29]. Kako bi bilo olakšano korištenje Adobe objektnog modela⁵, njihovih metoda i svojstava, uz Adobe paket dolazi aplikacija Adobe ExtendScript Toolkit. Aplikacija služi za pisanje i testiranje skripti te za istraživanje objektnog modela programa za koji se piše skripta [24]. U navedenoj aplikaciji moguće je pisati skripte za sve Adobe programe koji podržavaju skriptiranje.

Objektni model InDesigna sačinjen je od kolekcije objekata koji reprezentiraju program, njegove komponente, dokument, stranice dokumenta i elemente na stranicama. Tako npr. dokument sadrži stranice, a stranice sadrže dijelove stranica (tekstualni okviri, vektorski oblici i dr.). Tekstualni okviri sadrže tekstualne znakove, riječi, odlomke, usidrene okvire, tablice, grafičke okvire i sl. Grafički okviri sadrže slike, EPS ili PDF datoteke. Nabrojene stavke su objekti koji tvore InDesign publikaciju i s kojima se radi kada se pišu skripte. Objekti su u publikaciji organizirani u specifičnom poretku: okviri su na stranicama, a stranice su u dokumentu koji je unutar InDesign aplikacijskog objekta. Kada se govori o objektnom modelu ili hijerarhiji, govori se upravo o toj strukturi. Slika 1. prikazuje

⁵ Objektni model je kolekcija objekata i klasa preko kojih program može ispitivati i rukovati nekim servisom ili sustavom

organizaciju osnovne razine InDesign objektnog modela i hijerarhiju u njemu. Slika ne prikazuje cjelokupni model programa, već samo njegov dio.



Slika 1. InDesign objektni model

Izvor: http://cssdk.s3-website-us-east-1.amazonaws.com/sdk/1.0/docs/WebHelp/app_notes/id_obj_model.htm

Objekti imaju svojstva (*eng. attributes*) koja se zadaju na određene vrijednosti. Primjerice, svojstva tekstualnog objekta uključuju korišteni font, njegovu veličinu, razmak između redaka i dr. Objekti također imaju metode, tj. akcije koje mogu odraditi. Primjerice, objekt dokument ima metode za ispis, izvoženje ili spremanje.

Razumijevanje objektnog modela je ključ pronalaženja objekta s kojim se želi raditi i samog pisanja skripti. Najbolji vodič za skriptiranje u InDesignu je poznavanje samog programa što olakšava rad s objektima. Objekti su u InDesignu logički poredani u hijerarhiju prema uobičajenom načinu upotrebe programa i izbornika.

Problem koji se često ističe kod JavaScript programa, relativno je sporo vrijeme izvođenja. Razlozi tomu nalaze se u načinu izvršavanja koda (pomoću interpretera) i u dinamičkim svojstvima objekata koja se trebaju dohvaćati tijekom izvođenja. Poboljšanje svojstava izvođenja kroz optimizaciju napisanog koda važna je metoda kojom bi uz pomoć JavaScripta izradili kvalitetne programske dodatke. Zato što JavaScript izjava može izvršiti

mного strojnih instrukcija⁶, male promjene u izvornom JavaScript kodu mogu uvelike poboljšati performanse. Optimizacija koda može biti rađena na dvije razine. Za prvu razinu odgovoran je programer koji optimizira izvorni kod te će optimizacija ovisiti o primijenjenim optimizacijskim tehnikama. Druga razina optimizacije odnosi se na optimizaciju koja je ugrađena u programski prevodilac (interpreter) koji opet može primjenjivati optimizacijske tehnike prilikom izvođenja koda [30].

Za pisanje JavaScript koda poznate su smjernice [31] i trebalo bi ih se držati kako bi se izrađivao efikasan kod. S ExtendScriptom je drugačija situacija. ExtendScript upotrebljava JavaScript sintaksu, ali manipulira InDesign objektnim modelom, koristi nova svojstva i mogućnosti koje mijenjaju standardnu definiciju jezika. Ne postoji adekvatna literatura o optimizaciji ExtendScript koda čak ni od strane proizvođača Adobe System Incorporated, koja bi se mogla primijeniti tijekom pisanja koda. Nedostupne su i informacije o načinu rada Adobe ExtendScript interpretera i o ugrađenim optimizacijama koje se odvijaju tijekom interpretiranja koda.

Iz navedenih se razloga ovim radom žele odrediti optimizacijske tehnike pri pisanju ExtendScript koda za Adobe programe.

⁶ Strojni ili asemblerski jezik (*eng. machine language*) osnovni je jezik računala. Naredbe pisane u strojnom kodu jedine su koje računalo razumije i ne treba ih prevoditi kako bi ih računalo izvodilo.

3.2. XML i SVG

Upotreba XML sustava može donijeti veću fleksibilnost, višestruke izlazne opcije, pretraživanje i indeksiranje što teško ostvarujemo bez upotrebe web tehnologija [32]. Zbog toga se razvijaju novi načini implementiranja ovakvih tehnologija, novi modeli rada [33], u već postojeće sustave, bez velikog ulaganja u nove resurse i s jednostavnom primjenom kako bi se novi modeli mogli lakše integrirati.

Za potrebe ovog istraživanja korišten je SVG skriptni jezik za vektorsku grafiku baziran na XML tehnologiji, a rađena je i automatizacija prijeloma XML podataka kako bi se ostvarila varijabilna grafička priprema. Za razliku od tradicionalne statične vektorske grafike, SVG nudi nove mogućnosti interaktivnosti, dinamičnosti prikaza, mogućnosti pretraživanja sadržaja kao i automatizacije više procesa preko svog objektnog modela [34] [35]. U usporedbi s rasterskom grafikom daje mogućnosti skaliranja i manje veličine datoteke [36]. SVG je tekstualni, skriptni format otvoren za upotrebu tako da se može lako dobavljati te učiti iz postojećih primjera [37]. SVG razdvaja dizajnirani prikaz i aplikacijsku logiku od sadržaja te se sadržaj može neovisno uređivati i pretraživati. Iz tog razloga dobivamo dinamički sustav za upravljanje sadržajem koji se odlikuje u redukciji troškova održavanja, vremenu uređivanja i povećanju pouzdanosti i fleksibilnosti sustava [38]. Grafike bazirane na SVG tehnologiji su efikasne, jednostavne za razvoj i održavanje te nude brojne mogućnosti vizualnog dizajna, organizacije podataka, animacije i interaktivnosti s korisnikom [39] što nam omogućuje unaprjeđivanje nekih postojećih modela digitalne grafičke pripreme.

Istraživanje novih tehnologija i načina njihove implementacije izuzetno je važno kako bi se poboljšali i unaprijedili postojeći radni procesi. Nove tehnologije se uvode na temelju eksperimentiranja, simulacijom, testiranjem i mjerenjima u praksi. Očekuje se da će se mnoge faze automatizirati i poboljšati XML derivatima. Upotreba SVG, VML i XSL-FO tehnologija sve više raste, stoga je potrebno napraviti korake u njihovoj implementaciji u grafičku pripremu s ciljem poboljšanja operacija, radnih procesa i navika [40].

4. EKSPERIMENTALNI DIO

Cilj eksperimentalnog dijela je istražiti mogućnosti optimizacije u proizvodnom procesu grafičke pripreme primjenom skriptiranja. Pretpostavka je da će napisane skripte donijeti veće uštede vremena prilikom prijeloma publikacija koje su ovisne o radnjama koje skripta zamjenjuje. Manje uštede se očekuju kod radnji koje vremenski kraće traju.

4.1. Metode rada i eksperimentalni okvir

U praksi se mogu identificirati dvije vrste zadataka grafičke pripreme. Zadatci generalne namjene, gdje je moguće jednom razvijen algoritam primijeniti na mnoge slučajeve jer problemi dijele slična svojstva, i specifične namjene gdje je područje primjene vrlo specifično i nije prenosivo na druga područja [41]. Pri izradi skripti za automatizaciju razmatrat će se oba pristupa. Birat će se zadatci koji se učestalo ponavljaju ili specifični zadatci koji vremenski dulje traju i zbog toga ih je poželjno automatizirati.

Planom istraživanja određene su sljedeće faze:

- I. Analiza problema i analiza izvedivosti pojedinog modela u okviru grafičke pripreme;
- II. Izrada prijedloga modela za izvođenje unaprjeđenja uz pomoć skriptnih jezika i njihovo implementiranje u definirani proces grafičke pripreme;
- III. Mjerenje unaprjeđenja i obrada mjernih rezultata;
- IV. Izvođenje zaključaka na temelju prikupljenih i obrađenih rezultata.

Za potrebe istraživanja korišteni su standardni alati u grafičkoj pripremi. Izabran je Adobe Design and Web Premium CS6 paket, a iz paketa su izabrani programi Adobe InDesign CS6 verzije 8.0.1., Adobe Illustrator CS6 i Adobe Photoshop CS6. Od skriptnih tehnologija korišteni su JavaScript, XML i SVG. Kako bi se lakše pretraživao Adobe objektni model, metode i svojstva objekata te kako bi olakšali testiranje skripti [24] korišten je Adobe ExtendScript Toolkit CS6 verzije 3.8.0.12., DOM (*Document Object Model*) verzije 8.0 i ExtendScript verzije 4.2.12.

Karakteristike računala na kojemu su rađena vremenska mjerenja su sljedeće:

- Operativni sustav: Windows 7 Professional, 64 bitni
- Procesor: Intel Core™ 2 Quad CPU Q8200, 2x2,33 GHz
- RAM: 4,00 GB.

4.2. Analiza problema i analiza izvedivosti pojedinog modela u okviru grafičke pripreme

Prva faza izrade istraživanja bila je detektiranje potencijalnih procesa za optimizaciju skriptiranjem. Navedena faza izravno je vezana s prvom hipotezom istraživanja kojom se pretpostavlja da je moguće temeljem analize segmenata proizvodnih procesa grafičke pripreme upotrebom skriptnih jezika pod određenim uvjetima grafičku pripremu optimalno ubrzati s povećanjem pouzdanosti i isplativosti.

Cilj analize je istražiti segmente proizvodnih procesa grafičke pripreme u kojima bi se mogla ostvariti optimizacija procesa upotrebom skriptnih jezika. Pretpostavka je da će napisana skripta donijeti uštedu u vremenu prilikom prijeloma publikacije koja je ovisna o radnjama koje skripta zamjenjuje. Iz analize je potrebno identificirati procese koje bi mogli poboljšati upotrebom skriptnih tehnologija. Promatrana je digitalna grafička priprema od pripremanja dostavljenih materijala za prijelom, izrade prijeloma, pa do izrade konačne montaže spremne za tisak. Analizom je bilo potrebno odrediti samo one postupke koje je moguće unaprijediti do definirane razine.

4.2.1. Kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju

Pri procjeni pogodnosti procesa za automatizaciju određeni su kriteriji odabira prikazani u Tablici 1.

Tablica 1. Kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju

KRITERIJ	Opis
1.	Proces je vremenski zahtjevniji (traje dulje od 30 min) i/ili sadrži mnogo slijednih koraka. Izrađena automatizacija može se primjenjivati i na druge slične procese. Upotrebom skripte povećava se pouzdanost prijeloma (manja je mogućnost pogreške pri oblikovanju sadržaja).
2.	Dizajn je baziran na pravilima ili predlošcima, s jednostavnim ili ponavljajućim prijelomom. Materijal koji se prelama sastoji se od mnogo stranica ili elemenata koje je potrebno ručno uređivati.
3.	Materijal se često objavljuje s podacima, tj. sadržajem ili dijelovima sadržaja koji dolaze iz baze podataka, XML dokumenta ili je izrađen pomoću skripte. Posao se odnosi na personalizaciju publikacije i varijabilni tisak.
4.	Novi model rada koji je moguće implementirati u programe za DTP kako bi se izbjegla dodatna ulaganja u nove alate.

Prvi kriterij odabran je jer se manja ušteda očekuje kod radnji koje vremenski kraće traju ili zahtijevaju manje slijednih koraka[4]. Posao koji zadovoljava navedeni kriterij je dugotrajan ili ponavljajući, zamoran da bi se odradio ručno [21]. Drugi kriterij određuje vrstu posla pogodnu za automatizaciju jer se odnosi na poslove koji se ne moraju ponavljati često, ali su dugotrajni za pripremu. Treći i četvrti kriterij odnose se na poslove koji bi mogli ostvariti nove dodane vrijednosti ili ubrzati procese personalizacije bez dodatnih ulaganja u nove alate.

Kako bi proces bio odabran za optimizaciju, mora zadovoljavati barem jedan od navedenih kriterija. Naime, neki se kriteriji međusobno isključuju. Nakon što su identificirani potencijalni procesi koje želimo unaprijediti, prije pisanja same skripte potrebno je izvesti analizu izvedivosti pojedinog modela i odrediti do kojeg će se nivoa izrađivati automatizacija. Pri izradi skripte potrebno je osigurati kriterij izvedivosti skripte u više radnih procesa jer je to jedan od čimbenika koji nam omogućuje postizanje optimizacije digitalne grafičke pripreme. Ukoliko izradimo skriptu koja je primjenjiva samo u vrlo rijetkim i specifičnim slučajevima, stvarna ušteda vremena i postignuta automatizacija mogu postati zanemarive [4]. Dakle, potrebno je odabrati procese koji će se unaprjeđivati tako da napisane skripte budu općenitije i primjenjive u različitim situacijama. To se može postići i uvođenjem promjenjivih parametara koje određuje korisnik skripte ili se određuju iz dokumenta koji se uređuje.

Prilikom analize procesa utvrđeni su i procesi koje je nepoželjno, teško ili nemoguće automatizirati. Pri tome su određeni sljedeći **kriteriji koji opisuju procese koji nisu podobni za optimizaciju** skriptiranjem:

1. Automatizacija zadataka koji nisu uobičajeni ili se rijetko ponavljaju;
2. Automatizacija prijeloma cjelokupnih periodičkih publikacija u kojima nije unaprijed određeno formatiranje dokumenta, nego je ovisno o odlukama dizajnera;
3. Automatizacija izrade složenih montaža.

Pod složenim montažama podrazumijevaju se montaže poslova s velikim brojem stranica, montaže knjiga za različite vrste uveza, upotreba različitih postavki montažnih araka u istom poslu i sl. Za složene montaže bolje je koristiti već dostupna standardna rješenja, uobičajena i testirana u grafičkoj pripremi kao npr. izrada montaže u programu Preps. U njemu se montaža složenih poslova brže odrađuje zbog prilagodbe programa samo navedenim zadatcima i zbog programskih jezika koji se brže izvršavaju od skriptnih. Kod skriptiranja takvi poslovi mogu dulje trajati.

4.2.2. Analiza procesa digitalne grafičke pripreme

Procesi grafičke pripreme bilježeni su u manjem grafičkom poduzeću Graphprint d.o.o. Osnovna djelatnost navedenog poduzeća je dizajn, prijelom, priprema za tisak te digitalni tisak i dorada manjih naklada. Na temelju njihovih radnih procesa željeli su se pronaći dijelovi stvarne grafičke pripreme koju je poželjno optimizirati. Analiza je rađena na način da su tijekom 3 mjeseca (63 radna dana) bilježeni poslovi vezani uz digitalnu grafičku pripremu koji su se ostvarivali u tom poduzeću. Bilježenje poslova radilo je troje zaposlenika. Tablica 2. prikazuje sumarna vremena u tom periodu.

Tablica 2. Vremena utrošena na radne zadatke grafičke pripreme u poduzeću Graphprint d.o.o. u periodu od 3 mjeseca (ožujak - svibanj 2013. godine)

	Opis zadataka	Utrošeno vrijeme (u satima)	Postotak utrošenog vremena	Mogućnost automatizacije (procjena djelatnika)
1.	Prijelom periodičkih publikacija	312	20,6 %	DJELOMIČNO/ DA
2.	Prijelom publikacija (neperiodičke publikacije, knjige, brošure, letci i sl.)	153	10,1 %	DJELOMIČNO/ DA
3.	Obrada fotografija (prebacivanje u CMYK sustav boja, postavljanje rezolucije, podešavanje krivulja, postavljanje oštine slike i dr.)	108	7,1 %	DJELOMIČNO/ DA
4.	Dizajn	69	4,6 %	DJELOMIČNO/ NE
5.	Montaža	48	3,2 %	DJELOMIČNO/ DA
6.	Priprema za tisak varijabilnih podataka (izrada numeriranih dokumenata, ubacivanje adresa, personalizacija navođenjem varijabilnih podataka, ubacivanje promjenjivih kodova)	36	2,4 %	DA
7.	Popravci dostavljenih materijala za tisak (dodavanje napusta, micanje sadržaja od linije reza, skaliranje dokumenta, mijenjanje sustava boje iz RGB u CMYK, brisanje neispravno postavljenih reznih oznaka, ispravci u tekstu)	29	1,9 %	DJELOMIČNO/ NE
8.	Ostali poslovi nevezani uz digitalnu grafičku pripremu	757	50,1 %	-
	UKUPNO	1512	100 %	-

U Tablici 2. poslovi su poredani po vremenu utrošenom za izradu i po zastupljenosti u svakodnevnim poslovima. Cilj analize bio je odrediti različite procese digitalne grafičke pripreme koji se primjenjuju u navedenoj tiskari i pronaći dijelove koje je moguće optimizirati skriptiranjem. Procjene djelatnika o mogućnostima automatizacije izrađene su prema unaprijed zadanim kriterijima (*4.2.1. Kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju*) koji su izvedeni iz teoretskih pretpostavki istraživanja.

4.3. Izrada prijedloga modela za izvođenje unaprijeđenja uz pomoć skriptnih jezika i njihovo implementiranje u definirani proces grafičke pripreme

Iz analize proizvodnih procesa te analize izvedivosti pojedinog modela utvrđeni su procesi poželjni za automatizaciju. Tablica 3. prikazuje odabrane procese i kriterije odabira koje udovoljavaju.

Tablica 3. Prijedlozi skripti za optimizaciju procesa digitalne grafičke pripreme

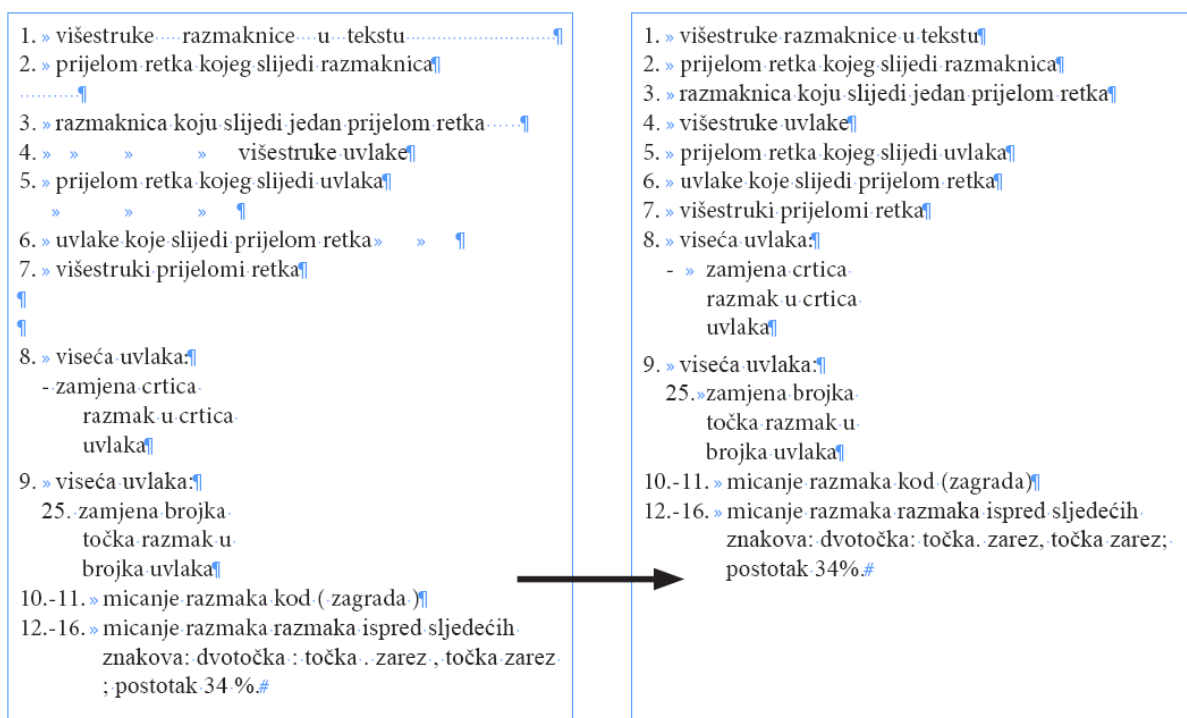
	Naziv prijedloga	Opis	Udovoljeni kriteriji (Tablica 1.)	Skripta automati-zira radne zadatke (Tablica 2.)	Predviđena vrsta automatizacije procesa grafičke pripreme
1.	Skripta za brisanje neželjenih bjelina u tekstu	Zadatak skripte je izbrisati sve neželjene bjeline u tekstu koji se prelama.	1., 2.	1., 2., 4.	Djelomična
2.	Skripta za zadržavanje uvezenih znakovnih stilova	Zadatak skripte je pronaći sve znakovne stilove koji se koriste u uvezenom dokumentu kako se oni ne bi izgubili prilikom primjene stilova odlomaka.	1., 2.	1., 2., 4.	Djelomična
3.	Skripta za uređivanje tablica u periodičkoj publikaciji	Zadatak skripte je ubrzati izradu prijeloma tablica u periodičkoj publikaciji. Zadatak prijeloma je dugotrajan, ponavljajući, s jednostavnim unaprijed određenim dizajnom.	1., 2.	1.	Djelomična
4.	Skripta za izradu numeracije i montaže kartica	Numeracija i montaža kartica radi se ručno u programu za prijelom. Zadatak skripte je automatizirati proces.	1., 2., 3., 4.	6.	Cjelovita
5.	Skripta za izradu personaliziranih dokumenata iz podataka u XML datoteci	Zadatak skripte je automatizirati procese rada s XML datotekama.	1., 2., 3., 4.	6.	Cjelovita

6.	Skripta za izradu jednostavnih montaža u programu za prijelom	Zadatak skripte je ubrzati montažu jednostavnih poslova u programu za prijelom.	1., 2., 3.	5.	Cjelovita
7.	Skripta za automatizaciju izrade ePub dokumenta koji sadrži SVG datoteke	Ne postoji mogućnost automatske izrade ePub dokumenta koji sadrže SVG datoteke. Zadatak skripte je omogućiti automatsku izradu i optimizaciju izvoženja ePub dokumenta.	1., 2., 4.	1., 2.	Djelomična
8.	Skripta za automatsku obradu fotografija	Zadatak skripte je automatizirati procese obrade fotografija u programu Adobe Photoshop.	1., 2.	3.	Cjelovita
9.	Skripta za izradu barkoda iz varijabilnih podataka	Barkod se ne može izrađivati automatski. Potrebno je uvoziti gotove kodove ili ulagati u nove alate kako bi se omogućila varijabilna grafička priprema. Zadatak skripte je omogućiti i automatizirati proces.	1., 2., 3., 4.	6.	Cjelovita
10.	Skripta za obilježavanje ispravaka u tekstu nakon lekture	Zadatak skripte je dodati novu mogućnost kojom će se u prelomljenom tekstu prikazivati izrađeni ispravci nakon lekture kako bi se omogućilo digitalno odobravanje izrađenih ispravaka u tekstu.	4.	1., 2., 7.	Cjelovita

Nakon detektiranja potencijalnih procesa za unaprjeđenje izrađeni su novi, poboljšani modeli rada. Pretpostavka je bila da će novi modeli biti jednostavniji u primjeni i donijeti unaprjeđenje procesa skraćivanjem vremena izrade, povećavanjem pouzdanosti sustava (smanjenjem broja pogrešaka) te povećanjem stupnja automatizacije (manji broj koraka u izradi ranije i nakon implementacije). Za testiranje je izrađeno svih 10 skripti.

4.3.1. Skripta za brisanje neželjenih bjelina u tekstu (oznaka: S1)

Izrada skripte odabrana je kako bi se skratili svakodnevni poslovi prijeloma teksta u InDesignu. Većina klijenata dostavlja tekst za prijelom kao Word dokument. Tekst koji se dostavi često ima pogreške u smislu neželjenih bjelina koje je potrebno izbrisati prije prijeloma. Poslovi koje obavlja skripta prikazani su na Slici 2.

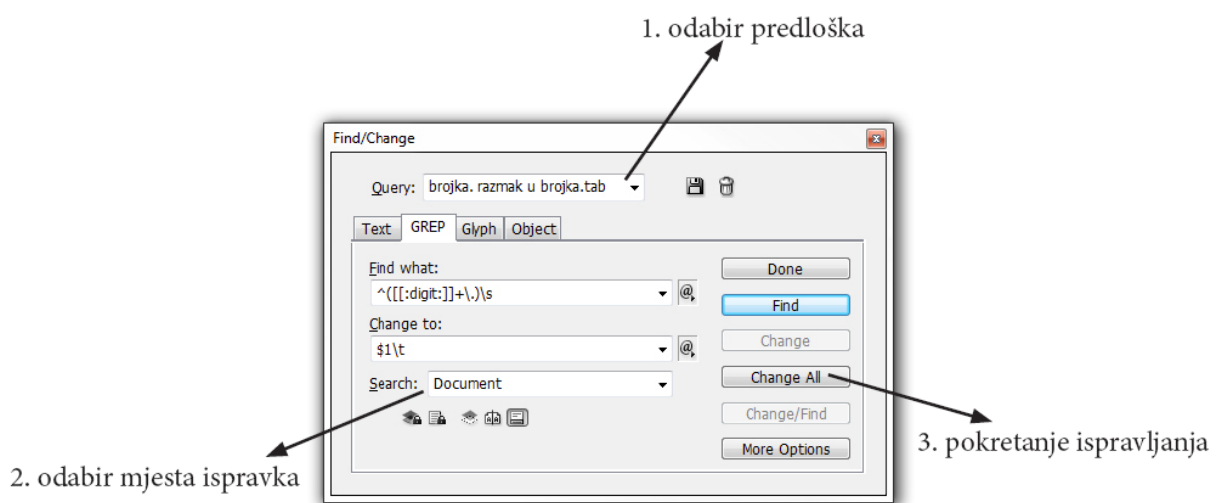


Slika 2. Poslovi brisanja neželjenih bjelina

Skripta služi za obavljanje sljedećih poslova u naznačenom redoslijedu:

1. traženje svih dvostrukih razmaknica i zamjena jednom razmaknicom,
2. traženje svih prijeloma retka koje slijedi jedna ili više razmaknica i zamjena jednim prijelomom retka,
3. traženje svih razmaknica koje slijedi jedan prijelom retka i zamjena prijelomom retka,
4. traženje svih višestrukih uvlaka (*eng. tab*) i zamjena jednom uvlakom,
5. traženje svih prijeloma retka koje slijedi jedna ili više uvlaka i zamjena jednim prijelomom retka,
6. traženje svih uvlaka koje slijedi prijelom retka i zamjena jednim prijelomom retka,
7. traženje svih višestrukih prijeloma redaka i zamjena jednim prijelomom retka,
8. zamjena svih crtica koje slijedi razmak na početku odlomka u crtica uvlaka,

9. zamjena svih brojaka na početku odlomka koje slijede točka i razmak u brojka, točka i uvlaka,
10. micanje razmaka nakon otvorene zagrade,
11. micanje razmaka prije zatvorene zagrade,
12. micanje razmaka ispred dvotočke,
13. micanje razmaka ispred znaka točka zarez,
14. micanje razmaka ispred znaka točke,
15. micanje razmaka ispred znaka zarez,
16. micanje razmaka ispred znaka postotka.



Slika 3. Poslovi brisanja neželjenih bjelina preko predložaka u *Find/Change* i GREP opcijama programa InDesign, bez korištenja skripte

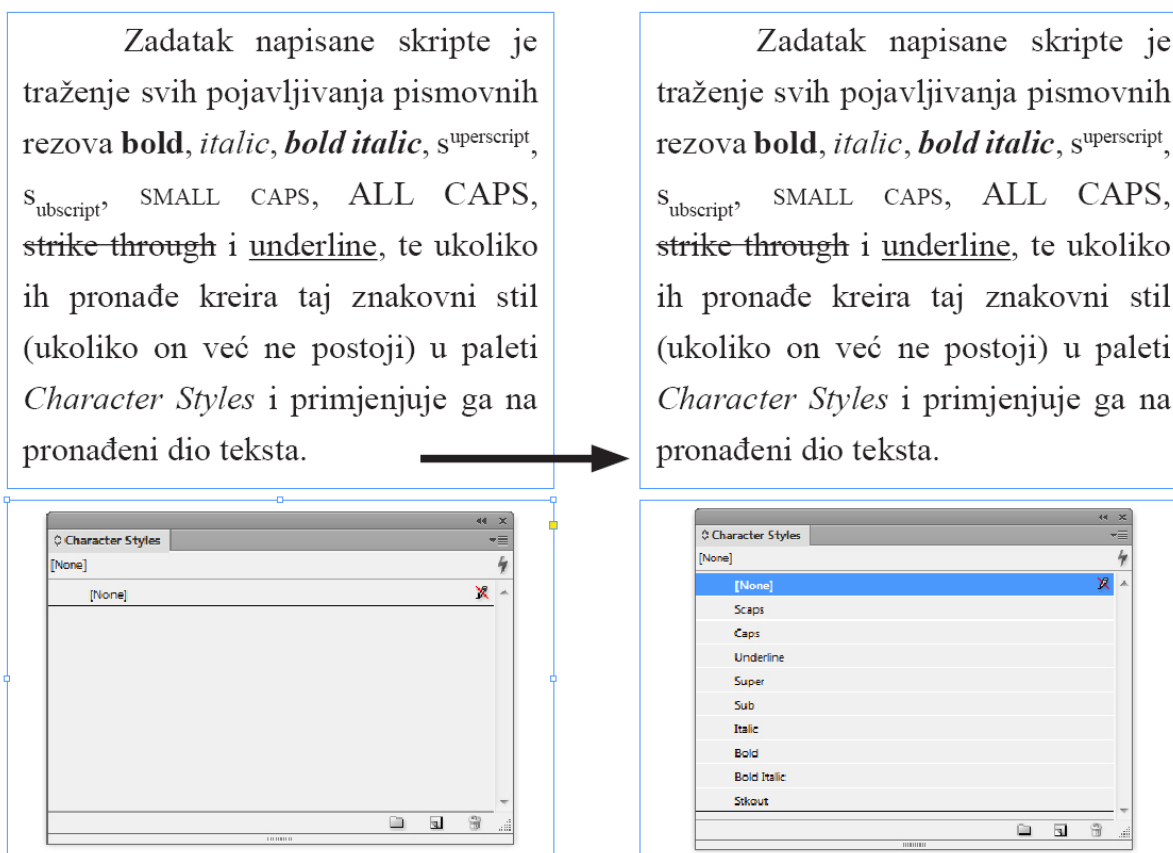
Svi navedeni postupci standardno su se obavljali jedan po jedan preko *Find/Change* i GREP⁷ opcija, spremljenih kao predložak u programu. Ukupno je nabrojeno 16 koraka koji su rađeni u navedenom redosljedu sa svakim dokumentom koji se prelamao. Slika 3. prikazuje način ranijeg obavljanja zadatka i navedene opcije u programu.

Izrađena skripta izvršava sve korake u zadanom redosljedu pokretanjem skripte te ne traži dodatnu interakciju s korisnikom. Varijabilnost skripte izrađena je samo u mogućnosti korisnika da odabere hoće li ispravci biti izrađeni u cijelom dokumentu ili samo u selektiranoj priči⁸.

⁷ GREP su regularni izrazi koji se koriste za pretraživanje teksta. To su izrazi koji su kriptični, ali izuzetno moćni u zadavanju uzoraka koji se traže u tekstu.

⁸ Priča (*eng. Story*) u InDesignu označava sadržaj svih okvira koji su međusobno povezani (*eng. linked*)

4.3.2. Skripta za zadržavanje uvezenih znakovnih stilova (oznaka: S2)



Slika 4. Automatizacija procesa postavljanja znakovnih stilova i izgled *Character Styles* palete nakon izvođenja skripte

Gotove knjige često dolaze lektorirane i spremne za prijelom kao Word dokument. U takvim dokumentima često su postavljeni znakovni stilovi koje je potrebno zadržati i u gotovom prijelomu. Pod znakovnim stilovima podrazumijevaju se stilovi koji nisu primijenjeni na cijeli odlomak, već samo na dio teksta u odlomku, a odnose se na pismovne rezove kao npr. kurziv (*eng. italic*) ili debela slova (*eng. bold*). Procedura bez upotrebe skripte sastojala se od izrade potrebnih stilova u *Character Styles* paleti, nakon čega je trebalo preko *Find/Change* predložaka pronaći navedena formatiranja i postaviti potreban znakovni stil. Tako izrađene promjene ne dozvoljavaju da se znakovni stilovi prilikom prijeloma u InDesignu izgube nakon primjene stilova odlomaka (*eng. Paragraph Styles*) što je čest slučaj ukoliko znakovni stilovi nisu primijenjeni. Ukoliko nisu postavljeni znakovni stilovi, a primjenjuju se stilovi odlomaka, često se gubi formatiranje slovnih znakova u odlomku koje je tada potrebno naknadno postavljati prema ispisu originalno dostavljenog dokumenta.

Zadatak skripte je traženje svih pojavljivanja pismovnih rezova *bold*, *italic*, *bolditalic*, *superscript*⁹, *subscript*¹⁰, *small caps*¹¹, *all caps*¹², *strike through*¹³ i *underline*¹⁴ te pri pronalasku traženog oblikovanja kreiranje znakovnog stila (ukoliko on već ne postoji). Novonastali stil dodaje se u paletu *Character Styles* i primjenjuje na pronađene dijelove teksta. Slika 4. prikazuje zadatke koji se obavljaju te izgled *Character Styles* palete nakon izvođenja zadataka.

⁹ Tekst postavljen kao eksponent

¹⁰ Tekst postavljen kao indeks

¹¹ Tekst prikazan verzalnim znakovima visine kurentnih slova

¹² Sva slova riječi predstavljena verzalnim znakovima

¹³ Tekst koji je precrtan

¹⁴ Tekst koji je podcrtan

4.3.3. Skripta za uređivanje tablica u periodičkoj publikaciji (oznaka: S3)

Snimanjem poslovnog procesa u poduzeću Graphprint d.o.o. primijećen je velik utrošak vremena pri grafičkoj pripremi Službenog glasnika Međimurske županije. Publikacija izlazi jednom ili dvaput mjesečno pa prema svojoj periodičnosti pripada u zadatke koji se učestalo ponavljaju. Dizajn joj je baziran na unaprijed zadanim pravilima. Većina utrošenog vremena pri prijelomu navedene publikacije trošila se na uređivanje i oblikovanje tablica proračuna.

Tablice proračuna koje se objavljuju u Glasniku dostavljaju se za objavu u digitalnom obliku kao Excel tablice. Sve općine koje objavljuju u Službenom glasniku Međimurske županije, koriste isti program za izradu Excel tablica proračuna za objavu. Prije samog prijeloma potrebno ih je dodatno urediti. Uređivanje obuhvaća postupke premještanja vrijednosti u odgovarajuće stupce, brisanje praznih redova i stupaca, brisanje datuma, brojeva stranica i vremena. Na Slici 5. prikazana je tablica koja se dobiva za prijelom.

	A	B	C	D	E	F	G	H	I	J	K	L	M
7													
8													
9		Proračun Općine Strahoninec za 2012 godinu sadrži kako slijedi:											
10		I. OPĆI DIO											
11		RAČUN PRIHODA I RASHODA											
12													
13		Konto	Naziv				Plan				Izmjene i dopune		
14		6	Prihodi poslovanja				8.002.745,00			570.731,43		8.573.476,43	
15		7	Prihodi				3.263.000,00			-1.585.750,00		1.677.250,00	
16		3	Rashodi poslovanja				2.163.745,00			-153.926,27		2.009.818,73	
17		4	Rashodi				8.850.000,00			-861.092,30		7.988.907,70	
18			Razlika - višak/manjak ((6 + 7) - (3 + 4))				252.000,00				0,00	252.000,00	
19		3. RAČUN FINANCIRANJA											
20													
21		Konto	Naziv				Plan				Izmjene i dopune		
22		5	Izdaci za financiranje				252.000,00				0,00	252.000,00	
23			Neto financiranje (8 - 5)				-252.000,00				0,00	-252.000,00	
24			Ukupno prihodi i primici				11.265.745,00			-1.015.018,57		10.250.726,43	
25			Ukupno rashodi i izdaci				11.265.745,00			-1.015.018,57		10.250.726,43	
26			Višak/Manjak + Neto financiranje						0,00		0,00		0,00
27													
28													
29													
30		RAČUN PRIHODA I RASHODA											
31													
32		Konto	Izvor	Naziv			Plan				Izmjene i dopune		
33		6	Prihodi poslovanja				8.002.745,00			570.731,43		8.573.476,43	
34		61	Prihodi od poreza				2.172.000,00			517.000,00		2.689.000,00	
35		611	Porez i prizrez na dohodak				2.000.000,00			500.000,00		2.500.000,00	
36		6111	011 Porez i prizrez na dohodak				2.000.000,00			500.000,00		2.500.000,00	
37		613	Porezi na imovinu				100.000,00			0,00		100.000,00	
38		6131	011 Stalni porezi				100.000,00			0,00		100.000,00	
39		614	Porezi na robu i usluge				65.000,00			17.000,00		82.000,00	
40		6142	011 Porez na promet				20.000,00			7.000,00		27.000,00	
41		6145	011 Porezi na imovinu				45.000,00			10.000,00		55.000,00	
42		616	Ostali prihodi od poreza				7.000,00			0,00		7.000,00	
43		6163	011 Ostali neraspoređeni prihodi od poreza				7.000,00			0,00		7.000,00	
44		63	Pomoći iz inozemstva				4.679.245,00			35.000,00		4.714.245,00	
45		633	Pomoći iz proračuna				4.679.245,00			35.000,00		4.714.245,00	
46		6331	011 Tekuće pomoći iz proračuna				15.000,00			35.000,00		50.000,00	
47		6332	011 Kapitalne pomoći iz proračuna				4.664.245,00			0,00		4.664.245,00	
48		64	Prihodi od imovine				172.500,00			7.731,43		180.231,43	
49		641	Prihodi od financijske imovine				3.000,00			4.731,43		7.731,43	

Slika 5. Tablica proračuna prije obrade; Excel dokument

Nakon obrade u Excelu tablice proračuna ubacuju se u InDesign gdje su potrebna brojna dodatna oblikovanja. Uvezena tablica prikazana je na Slici 6. Oblikovanje uključuje: podešavanje širine stupaca prema unaprijed definiranim vrijednostima, ovisno o broju stupaca

tablice, postavljanje stila tablice, postavljanje stila ćelija, postavljanje znakovnog stila, podešavanje visine redaka, isključivanje lomljenja riječi, brisanje prijeloma redaka unutar ćelija i zamjena razmaknicama (upotrebom *Find/Change* opcije), postavljanje udaljenosti teksta u ćelijama od rubova ćelija, dodavanje početnog retka u koji se upisuje „u kunama“, pretvaranje prva dva retka u naslovne retke koji se ponavljaju na svakoj stranici (*eng. header rows*), postavljanje stilova na prvi i drugi redak tablice. Nakon što su odrađene početne radnje, potrebno je izraditi najdulji dio posla: postavljanje posebnog stila na retke ovisno o sadržaju početne ćelije u svakom retku. Pri oblikovanju redaka brišu se prazni redci, ukoliko su oni uvezeni u dokument, te se spajaju ćelije kod redaka koji ne počinju brojčanim kontom. Postavljanje stilova radi se prečacima tipkovnice tako da se najprije selektira redak i nakon toga primijeni stil te po potrebi spoje ćelije.

14. prosinca 2011. "SLUŽBENI GLASNIK MEĐIMURSKE ŽUPANIJE" Broj 16 - Stranica 1729

Konto	Izvor	Naziv	Plan	povećanje smanjenje	Izmjene i popune
001A001			673.220,00	-68.967,57	604.252,43
			330.220,00	-29.066,57	301.153,43
			330.220,00	-29.066,57	301.153,43
01			330.220,00	-29.066,57	301.153,43
01	Rashodi	Rashodi	330.220,00	-29.066,57	301.153,43
01	Rashodi	Rashodi za poslovanje	324.220,00	-25.066,37	299.153,43
011	Rashodi	Rashodi za zaposlene	224.500,00	-27.296,62	197.203,38
0111	Rashodi	Rashodi za zaposlene (Bruto) Plaća za redovan rad	224.500,00	-27.296,62	197.203,38
01111	Rashodi	Plaća za zaposlene	186.500,00	0,00	186.500,00
01112	Rashodi	Plaća za zaposlene - veština	33.000,00	-31.000,00	2.000,00
01114	Rashodi	Plaća za zaposlene - redovan rad	5.000,00	3.703,38	1.296,62
012	Rashodi	Ostali rashodi za zaposlene	7.000,00	2.000,00	9.000,00
0121	Rashodi	Ostali rashodi za zaposlene	7.000,00	2.000,00	9.000,00
01211	Rashodi	Ostali rashodi za zaposlene - ostali	6.000,00	2.000,00	8.000,00
01213	Rashodi	Ostali rashodi za zaposlene - doprinosi	1.000,00	0,00	1.000,00
013	Doprinosi	Doprinosi	92.720,00	230,05	92.950,05
0131	Doprinosi	Doprinosi za mirovinsko osiguranje	49.035,00	230,05	49.265,05
01311	Doprinosi	Doprinosi za mirovinsko osiguranje	35.200,00	0,00	35.200,00
01312	Doprinosi	Doprinosi za mirovinsko osiguranje II	11.835,00	0,00	11.835,00
01313	Doprinosi	Doprinosi za mirovinsko osiguranje III	2.000,00	230,05	2.230,05
0132	Doprinosi	Doprinosi za obvezno osiguranje	38.535,00	0,00	38.535,00
01321	Doprinosi	Doprinosi za obvezno osiguranje	35.200,00	0,00	35.200,00

Slika 6. Uvezena tablica proračuna prije obrade u programu InDesign

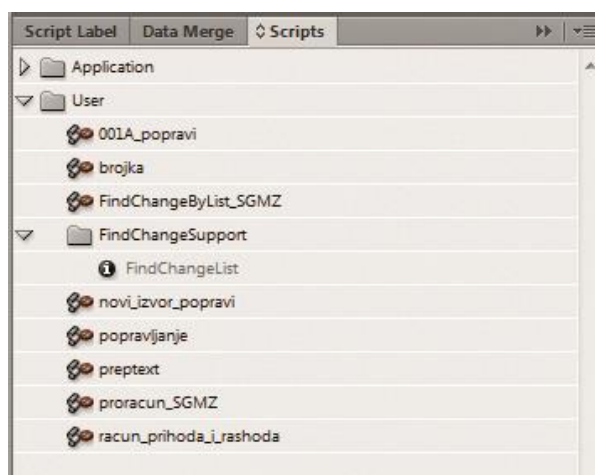
Također je potrebno napomenuti da se svaki proračun sastoji od 5 ili 6 posebnih tablica koje se nešto drugačije uređuju, ovisno o njihovom sadržaju i broju stupaca. Na Slici 7. prikazan je izgled prelomljenih tablica.

Stranica 2064 - Broj 17 "SLUŽBENI GLASNIK MEĐIMURSKE ŽUPANIJE" 21. prosinca 2012.					
Konto	Izvor	Naziv	1. izvjesje i dopuna	2. izvjesje i dopuna	3. izvjesje i dopuna
332	011	Rashodi za usluge	2.622.000,00	223.200,00	2.378.800,00
334	011	Naknade troškova osobama izvan radnog odnosa	20.000,00	22.000,00	42.000,00
339	011	Ostali nepremisljeni rashodi podrijetla	794.500,00	-12.500,00	782.000,00
34		FINANCIJSKI RASHODI	205.500,00	-54.700,00	240.200,00
342	011	Kamate za preuzeta kredita i zajmove	200.000,00	-80.000,00	120.000,00
343	011	Ostali financijski rashodi	25.500,00	-4.700,00	19.800,00
35		SUBVENCIJE	100.000,00	0,00	100.000,00
352		Subвенции и преносима друштвених, политичких, професионалних i obiteljnih izvan javnog odnosa	100.000,00	0,00	100.000,00
36		POMOĆI DANE U ZNOCIMITVO I UNUTAR OPĆEO PROBAČNA	0,00	70.000,00	70.000,00
363	011	Pomoći unutar općine proračuna	0,00	70.000,00	70.000,00
37		NAKNADE GRAĐANIMA I KUĆANSTVIMA NA IMENI OŠUGRAĐANI I DRUGE NAKNADE	388.000,00	-51.000,00	337.000,00
372	011	Ostale naknade građanima i kućanstvima iz proračuna	388.000,00	-51.000,00	337.000,00
38		OSTALI RASHODI	4.145.500,00	34.600,00	4.180.100,00
381	011	Dokuzi donacije	1.811.000,00	4.100,00	1.815.100,00
382	011	Kapitalne donacije	500.000,00	50.000,00	550.000,00
388	011	Iznovna rasloži	40.000,00	-30.000,00	10.000,00
389	011	Kapitalni prihodi	2.147.000,00	7.500,00	2.154.500,00
4		RASHODI ZA NABAVU NEFINANCIJSKE IMOVINE	6.873.100,00	871.400,00	7.744.500,00
41		RASHODI ZA NABAVU NEPROIZVEDNE DUGOTRAJNE IMOVINE	114.500,00	-80.000,00	34.500,00
411		Kapitalna imovina - prisvota bogatstva	114.500,00	-80.000,00	34.500,00
42		RASHODI ZA NABAVU PROIZVEDNE DUGOTRAJNE IMOVINE	6.558.600,00	651.400,00	7.209.000,00
421	011	Gradeništvo objekte	5.868.400,00	650.000,00	6.518.400,00
422	011	Postrojaja i oprema	247.400,00	-230.100,00	109.200,00
423	011	Prejuzna sredstva	20.000,00	-30.000,00	0,00
424	011	Kupnja, najam i ostala imovna sredstva	483.800,00	-27.400,00	211.000,00
425	011	Vingodiljiti naslozi i ostanak ostala	0,00	111.000,00	111.000,00
426	011	Nematerijalna prouzvedena imovina	147.000,00	-10.000,00	147.000,00
		Ukupno rashodi	11.780.000,00	676.800,00	12.456.800,00
B. RAČUN FINANCIRANJA					
Konto	Izvor	Naziv	1. izvjesje i dopuna	2. izvjesje i dopuna	3. izvjesje i dopuna
54		IZDACI ZA FINANCIRANJE DIOVITNI I OTPLATE RAZNOVA	800.000,00	0,00	800.000,00
544		OTPLATE POMOĆI U ZNOCIMITVO I ZAJMOVA od izvedbi i ostalih finansijskih institucija izvan javnog odnosa	800.000,00	0,00	800.000,00
		Ukupno izdaci	800.000,00	0,00	800.000,00

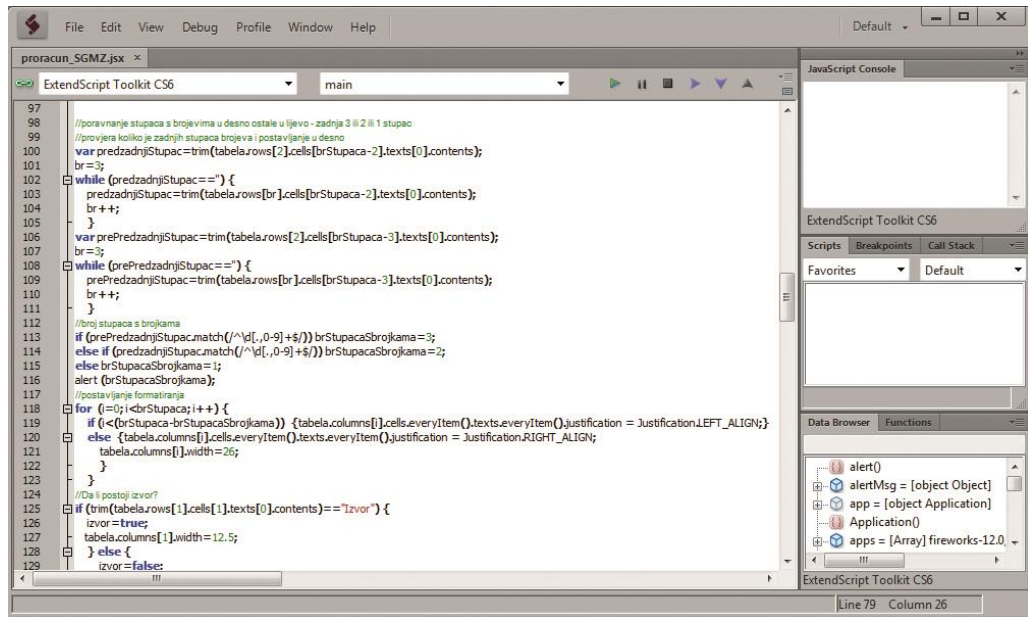
Slika 7. Uređena tablica proračuna u programu InDesign

Zbog navedenog formatiranja, prijelom ovakvih tablica je dugotrajan te postoji velika mogućnost pogreške zbog čestih izmjena stilova i ponavljajućih radnji u primjeni odgovarajućeg stila.

Za prijelom pomoću skripte izrađeno je nekoliko skripti koje se uzastopce primjenjuju ovisno o potrebi. Naime, Excel datoteke se razlikuju ovisno o vrsti proračuna koji se ispisuje, npr. polugodišnje izvješće o izvršenju, plan proračuna ili rebalans proračuna. Izrađene skripte prikazane su na Slici 8., a na Slici 9. djelomični pregled jedne od skripti u alatu ExtendScript Toolkit CS6.

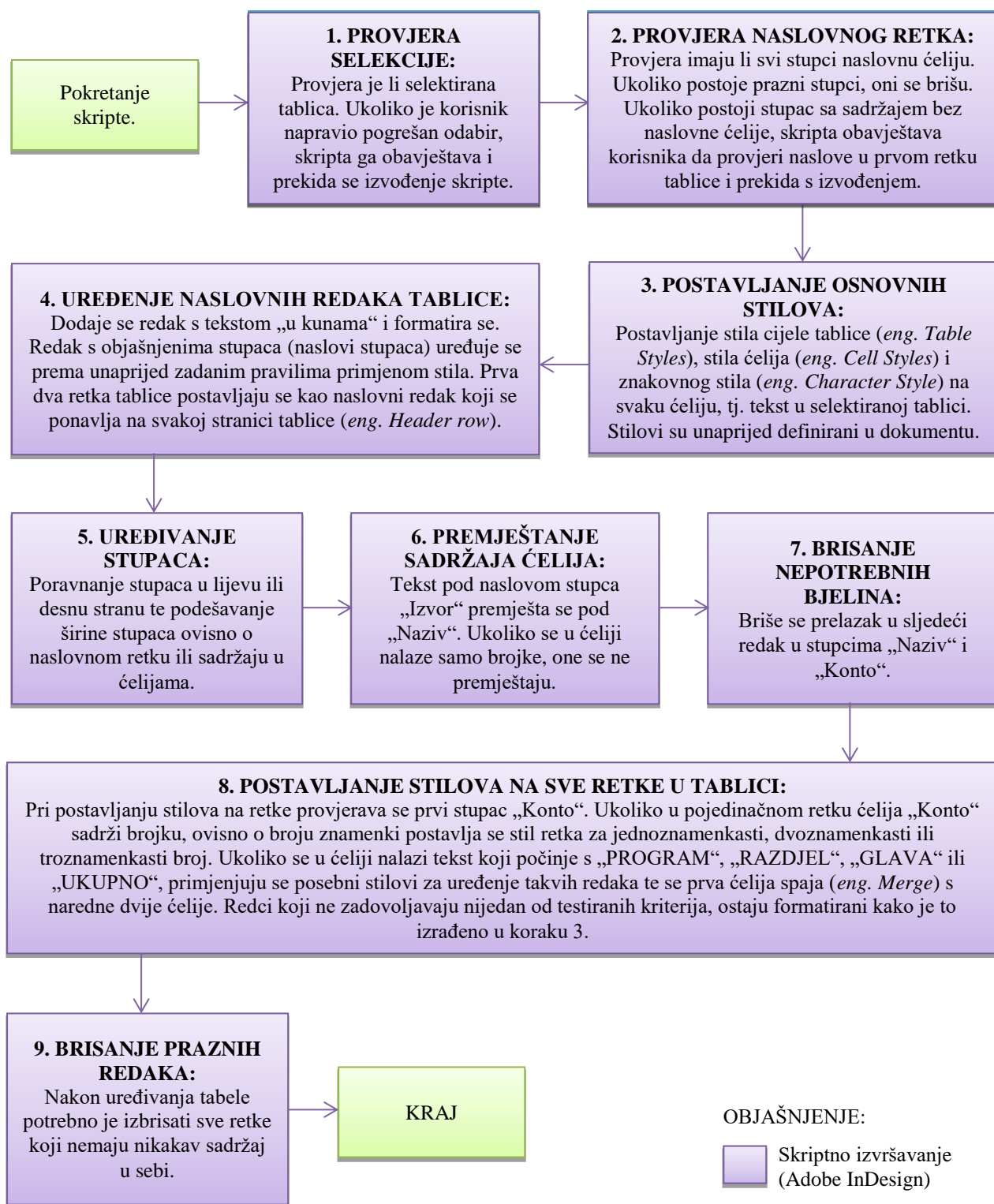


Slika 8. Izrađene skripte za prijelom proračuna za Službeni glasnik Međimurske županije



Slika 9. Dio skripte u alatu ExtendScript Toolkit CS6

Kako bi se izradio prijelom pomoću skripti, u Excelu je jedino potrebno napraviti poravnanje naslovnog retka i novčanih vrijednosti u ispravne stupce ukoliko je to potrebno. Nakon toga svih se pet tablica pojedinačno ubacuje u program te se primjenjuju odgovarajuće skripte. Jedino što korisnik mora napraviti prije primjene skripte, je provjeriti ima li svaki stupac odgovarajući naziv u prvoj ćeliji stupca. Iz naslova stupaca u prvom retku skriptno se određuje širina stupca. Skripta upozorava ukoliko neki naziv nedostaje. Nakon prijeloma potrebno je provjeriti zahtijevaju li tablice dodatno podešavanje. Ako je potrebno, i za to su napisane posebne skripte. Cijelo formatiranje, slaganje, spajanje ćelija i premještanje teksta uključeno je u navedene skripte. Slika 10. prikazuje tijek izvođenja napisane skripte koja mijenja operatera u izvršavanju zadatka.



Slika 10. Tijek izvođenja skripte kod prijeloma proračuna za objavu u Službenom glasniku Međimurske županije

4.3.4. Skripta za izradu numeracije i montaže kartica (oznaka: S4)

Varijabilni ispis podataka, poznat kao VDP (*eng. Variable Data Printing*), oblik je digitalnog tiska gdje se u samom tijeku tiska između svakog otiska mogu mijenjati podatci poput teksta, slika i grafika bez da se zaustavlja ili usporava proces tiska. Ovakav oblik tiska koristi se u izravnom marketingu gdje se otiskuju personalizirana pisma, brošure ili računi. Kod varijabilnog tiska postoji predložak, osnovni izgled potrebnog dokumenta, na koji se dodaju ostali podatci koji se vuku iz baze podataka ili vanjske datoteke [42].

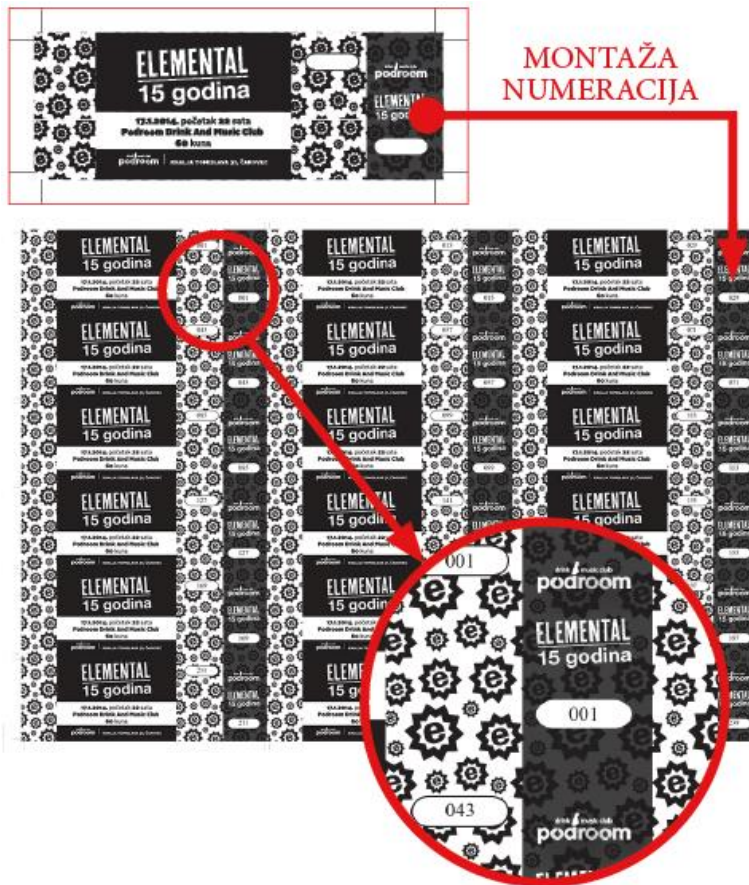
Pri analizi proizvodnih procesa nekoliko puta su se pojavili zadatci izrade montaže i numeracije različitih kartica kao npr. ulaznica, kupona, bonova, blokova i sličnih proizvoda. Poduzeće koje je analizirano ne posjeduje poseban program za izradu i ispis varijabilne grafičke pripreme. Takvi zadatci rješavani su uz pomoć programa Adobe InDesign i Excel.

Tijek standardnog proizvodnog procesa koji služi za izradu numeriranih i montiranih kartica izvršavao se kako slijedi:

Na gotov predložak, koji se uvozi kao PDF datoteka u program Adobe InDesign, postavlja se jedan ili više tekstualnih okvira koji će sadržavati varijabilne podatke. Nakon pozicioniranja i kreiranja stilova numeracije izrađuje se montaža predloška i tekstualnih okvira pozicioniranjem kartica na *Master*¹⁵ stranicama dokumenta. Nakon montaže se u programu Excel izrađuju stupci s brojevima koji će se postaviti po karticama. Izrađeni brojevi kopiraju se u InDesign. Sadržaj se ubacuje u okvire, a okviri međusobno povezuju kako bi se dobio automatski tijek teksta.

Kako bi se navedeni proces automatizirao i olakšao, izrađena je skripta koja automatizira izradu varijabilne grafičke pripreme. Automatizirani su procesi ubacivanja kartice, izrade montaže te numeracije kartica kako bi se dobila priprema spremna za digitalni tisak. Konačni proizvod standardne i skriptne izrade je niz stranica u InDesign dokumentu koje imaju primijenjen montirani predložak s *Master* stranice. U ranije izrađenim tekstualnim okvirima nalazi se numeracija koja se mijenja po stranicama. Slika 11. prikazuje gotovu montažu i numeraciju koja se može dobiti izvođenjem skripte ili standardnom izradom.

¹⁵*Master* stranice u programu Adobe InDesign određuju izgled stranica na koje su primijenjene



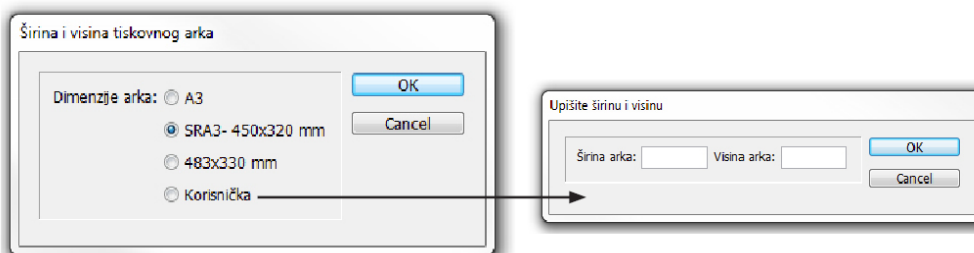
Slika 11. Montaža i numeracija uvezene datoteke

Skripta se izvršava kako slijedi:

1. Određivanje veličine arka za montažu.

Određuje se veličina stranice InDesign dokumenta koji se automatski otvara.

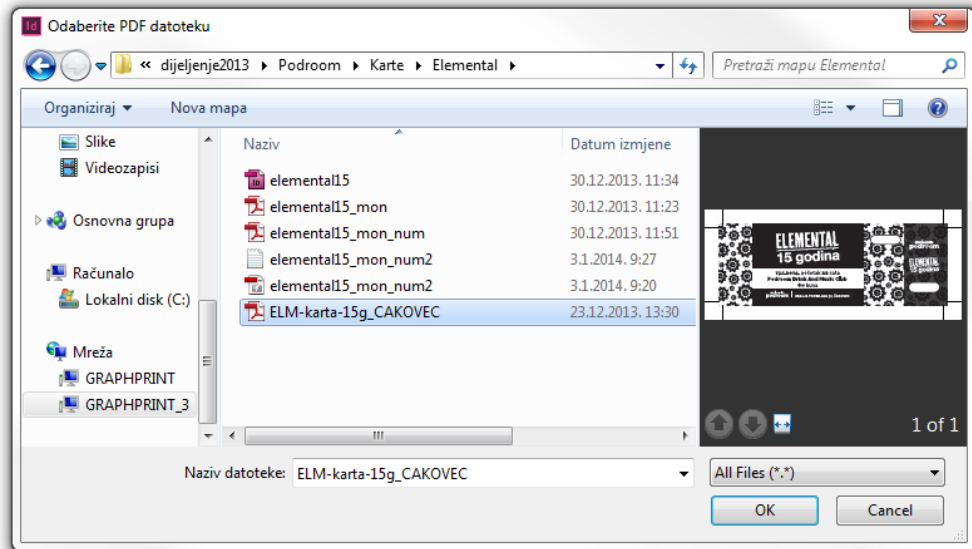
Izbornici za odabir prikazani su na Slici 12.



Slika 12. Određivanje veličine arka za montažu

2. Određivanje PDF dokumenta koji će se montirati.

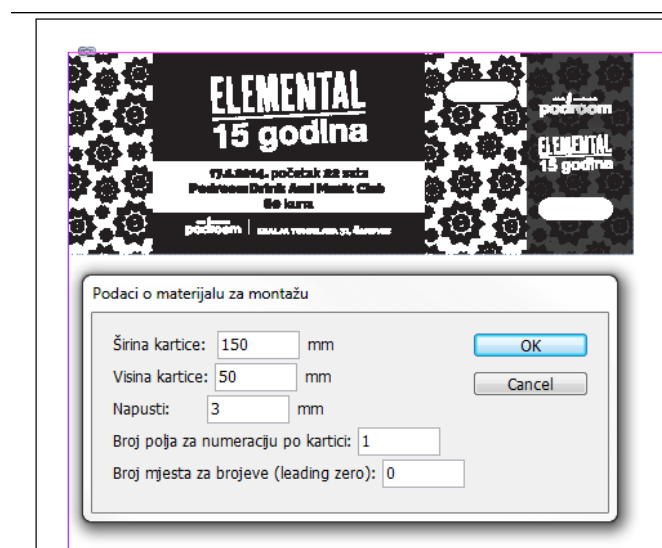
Slika 13. Prikazuje dijaloški okvir za odabir kartice za montažu.



Slika 13. Odabir PDF dokumenta za montažu

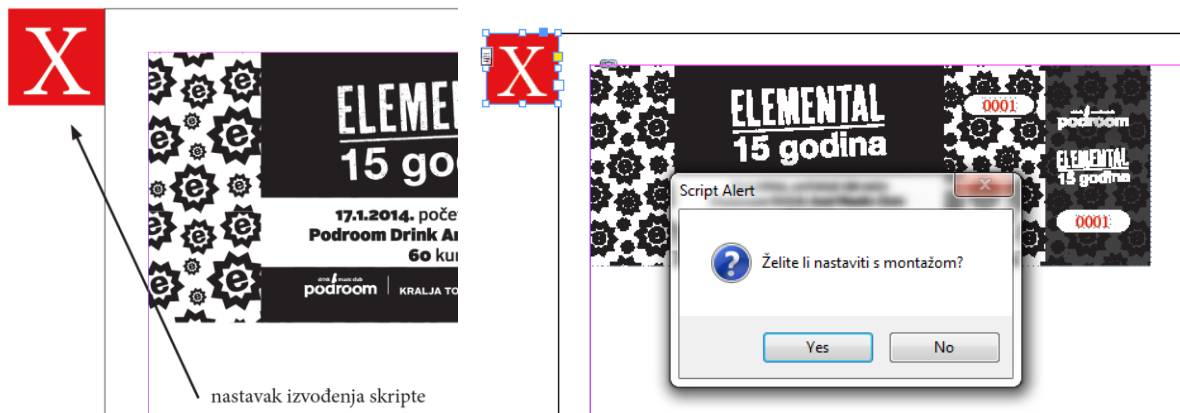
3. Ubacivanje PDF dokumenta u program, određivanje postavki montaže i numeracije.

Nakon ubacivanja dokumenta od korisnika se traži, ukoliko je potrebno, podešavanje širine, visine i veličine napusta kartice koja je ubačena. Također se određuje broj polja za numeraciju po kartici i oblik broja (broj mjesta za vodeće nule (*eng. leadingzero*)). Slika 14. prikazuje dijaloški okvir odabira navedenih postavki.



Slika 14. Određivanje postavki montaže i numeracije

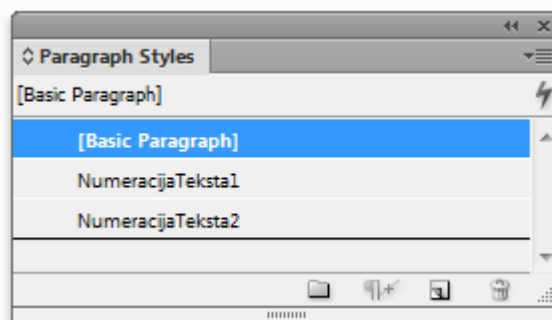
4. **Pozicioniranje i stiliziranje ubačenih tekstualnih okvira** za brojeve od strane korisnika te odabir nastavka izvođenja skripte. Slika 15. prikazuje gumb za nastavak izvođenja skripte.



Slika 15. Gumb za nastavak izvođenja skripte

5. **Odabir načina montiranja iz ponuđenih opcija.**

Skripta izračunava broj kartica na ranije selektiranoj veličini arka, sa zakretanjem i bez zakretanja kartice. Prije same montaže kreirat će se stilovi odlomaka koji će se primijeniti na brojeve koji se kasnije ubacuju. Stilovi odlomka se izrađuju kako bi kasnije bilo moguće, ukoliko se za to ukaže potreba, izmijeniti izgled ubačene numeracije. Slika 16. prikazuje stilove koje je kreirala skripta, a Slika 17. dijaloški okvir za odabir opcije montaže. Nakon odabira opcije montaže skripta izvodi montažu kartica te dodaje rezne oznake na gotovu montažu pozicioniranu na *Master* stranici.



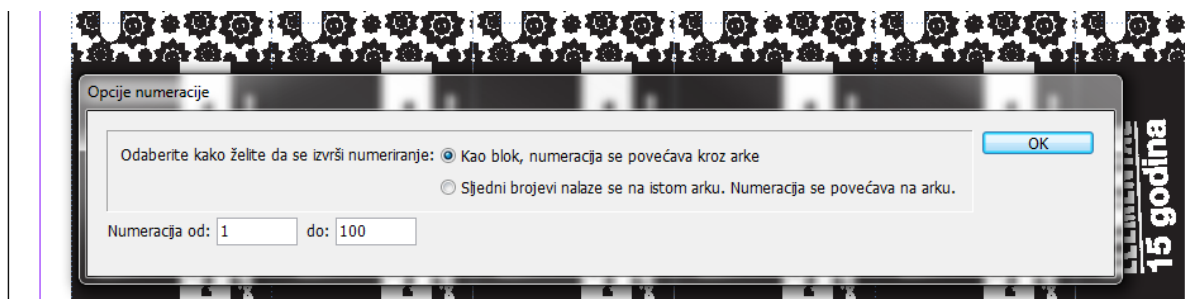
Slika 16. Stilovi koji su se kreirali iz ubačenih tekstualnih okvira za brojeve



Slika 17. Dijaloški okvir za odabir opcije montaže

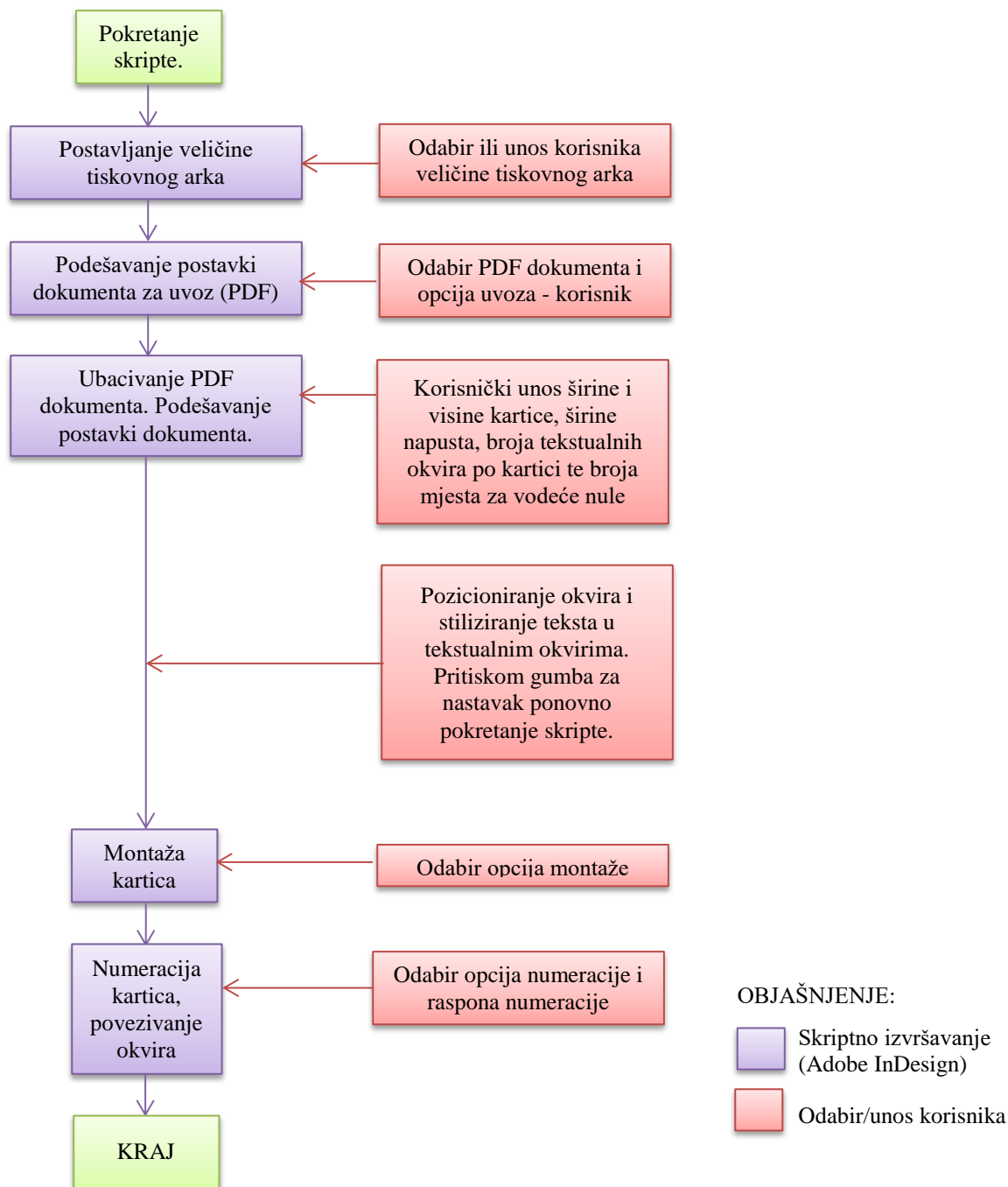
6. Određivanje postavki numeracije.

Slika 18. prikazuje izbornik za odabir opcija numeriranja. Opcije numeracije omogućuju odabir načina numeriranja i raspon numeracije. Opcija „Kao blok...“ omogućuje numeraciju kroz arke kako bi se nakon tiska i rezanja dobili blokovi sa slijednim brojevima u istom bloku, bez potrebe sortiranja. Opcija „Slijedni brojevi se nalaze na istom arku...“ numerira kartice slijedno na istoj stranici. Nakon potvrđivanja odabira skripta vrši numeraciju.



Slika 18. Izbornik za odabir opcija numeriranja

Slika 19. prikazuje tijek izvođenja skripte koja uključuje izradu numeracije i montaže kartica. Na slici je istaknuta interakcija korisnika i skripte te zadatci koje skripta izvršava.



Slika 19. Tijek izvođenja skripte za izradu numeracije i montaže kartica

4.3.5. Skripta za izradu montaže i personalizacije kartica iz podataka u XML datoteci (oznaka: S5)

U posljednje se vrijeme sve češće u tiskare dostavljaju XML datoteke koje se koriste za izradu grafičke pripreme. Upravo zbog zahtjeva klijenata potrebna je prilagodba novim načinima dostave materijala.

XML datoteke nastaju iz različitih sustava i programa i služe za razmjenu podataka između sustava. Glavne prednosti ovakvih datoteka su u tome što oznake (*eng. Tags*) dobro opisuju sadržaj, tj. podatci sami sebe dobro objašnjavaju. Podatke je moguće manipulirati pomoću standardnih alata i vrlo je lako kreirati različite preglede podataka upotrebom predložaka (*eng. Style sheets*). Za XML se mogu izrađivati i vlastiti alati jer je podržan u svim glavnim programskim jezicima uključujući C, C#, C++, Java, Perl, Python, Haskell, JavaScript i mnoge druge [43].

XML tehnologija omogućuje kreiranje različitih formata i medija iz istog izvora. XML nudi mogućnosti izrade publikacija u različitim verzijama izdanja: od tiskane knjige u promjenjivim formatima i s promjenjivim dizajnom do digitalnih izdanja na webu ili prijenosnim uređajima [44].

U grafičkoj se pripremi razmjena podataka uz pomoć XML-a najčešće odvija u slučajevima prijeloma podataka koji dolaze iz web sučelja. Dokumenti koji dolaze na prijelom kao XML datoteka mogu biti izrađeni i u standardnim programima kao što su primjerice Word ili Excel. Novije verzije navedenih programa i same koriste XML za pohranu sadržaja kako bi se njihovi podatci mogli lakše dijeliti. Može se kazati da je korištenje XML-a u računalnim sustavima postao standard za razmjenu podataka, a sve više ulazi i u digitalnu grafičku pripremu.

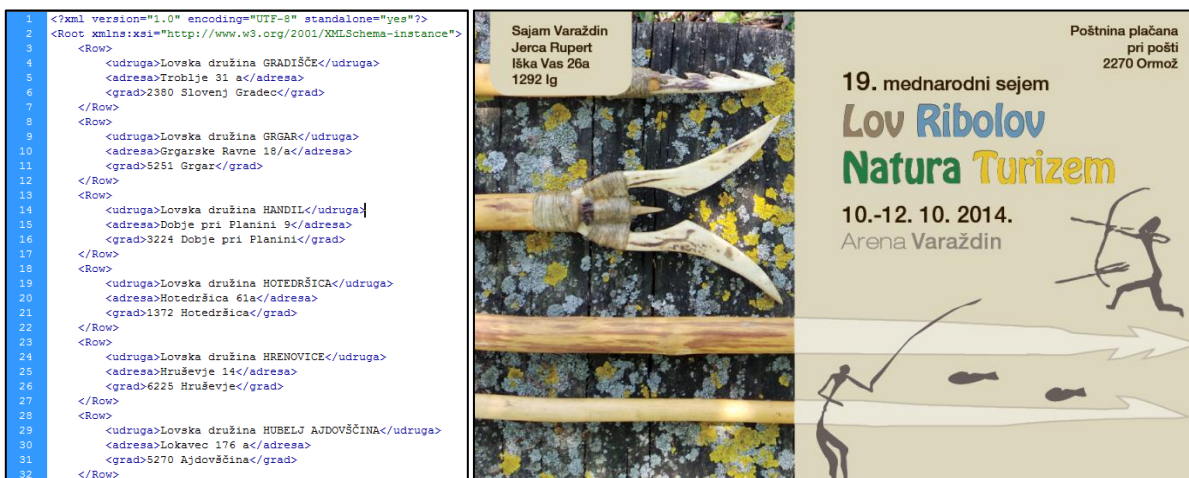
InDesign podržava rad s XML-om pomoću već ugrađenih paleta i izbornika. U programu je omogućeno uvoženje XML podataka i njihova upotreba. XML dokumenti mogu se koristiti kao baza sadržaja dokumenta. Pri radu s XML podacima uobičajeno se koriste dvije palete za izradu prijeloma: *Tags* i *Structure* [45].

Standardni način rada s XML dokumentima koji se želio unaprijediti uobičajeno se odvijao na sljedeći način:

- Najprije se na *Master* stranici kreira dizajn koji će se ponavljati na svim stranicama. Dizajn sadrži grafike i tekstualne okvire u koje se kasnije ubacuje sadržaj. Nakon izrade dizajna izrađuje se montaža i postavljaju rezne oznake.

- Proces rada sa XML-om započinje uvoženjem oznaka iz XML dokumenta u *Tags* paletu i povezivanjem oznaka sa stilovima odlomka ili znakova (eng. *Character* i *Paragraph Styles*).
- Kreira se izgled teksta po stranicama i povezuje sa oznakama. Na taj način kreira se struktura dokumenta u *Structure* paleti koja mora odgovarati strukturi XML dokumenta koji će se kasnije uvesti.
- Nakon što su svi tekstualni okviri i oznake postavljeni (izrađen je predložak za uvoz XML dokumenta), uvozi se XML koji se automatski postavlja po stranicama dokumenta preko *Autoflow* opcije tekstualnih okvira. Ubacivanje svih elemenata postiže se povlačenjem *root* elementa iz *Structure* palete na tekstualni okvir u kojemu se nalaze oznake XML elemenata.

Kako navedeni postupak iziskuje mnogo slijednih koraka, isti se želio pojednostaviti korištenjem skriptiranja. Slika 20. prikazuje ulazne dokumente, a Slika 21. rezultat skripte koja se testirala.



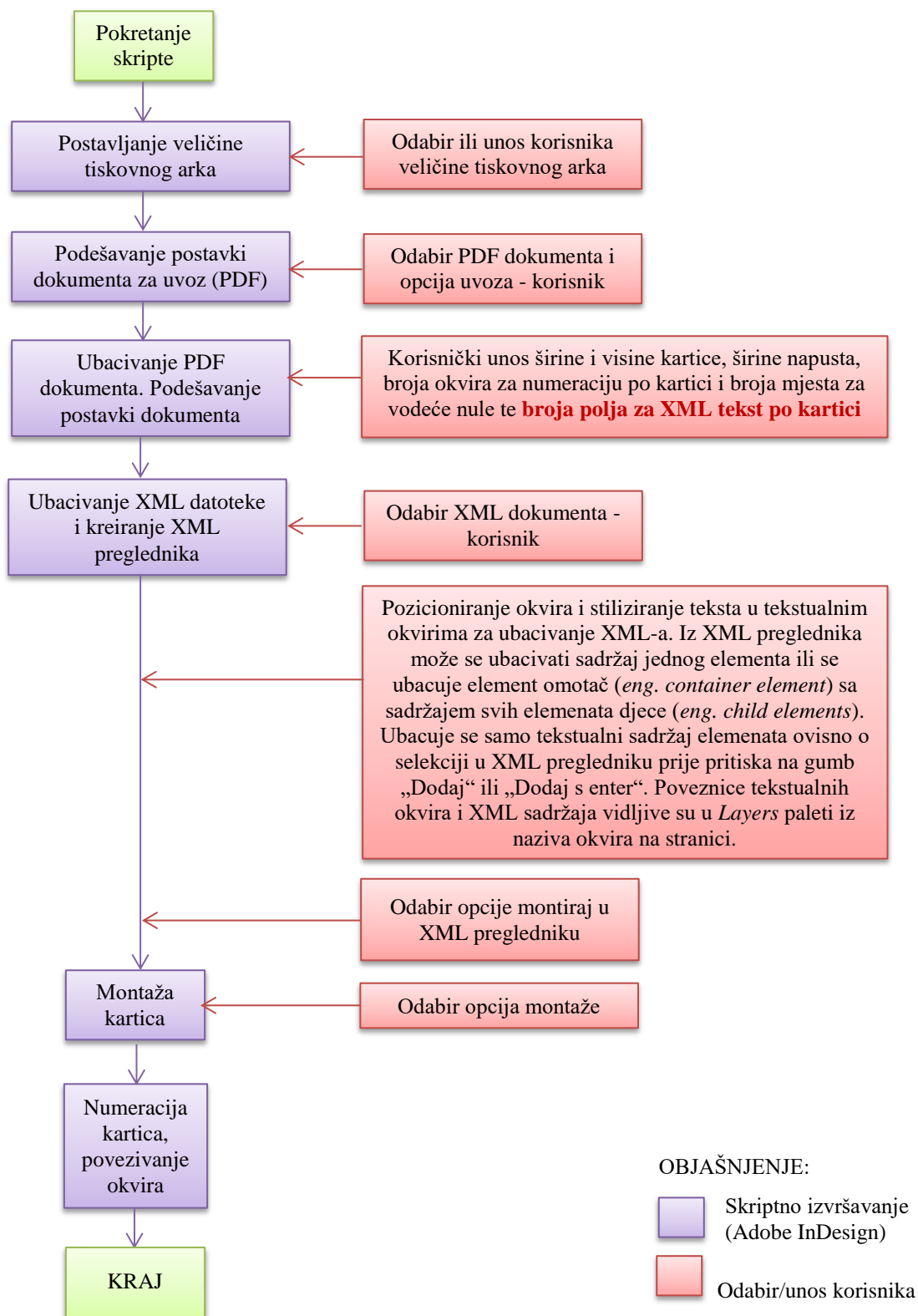
Slika 20. Ulazni dokumenti za skriptu: dio XML datoteke za prijelom i kartica za montažu



Slika 21. Konačni rezultat skripte; montirane kartice s raspoređenim i oblikovanim XML podatcima

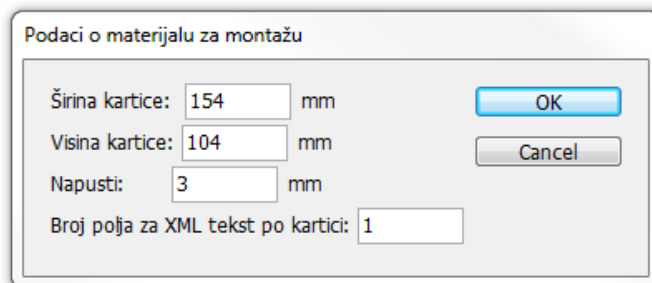
Dijelovi skripte preuzeti su iz prethodne skripte 4.3.4. *Skripta za izradu numeracije i montaže kartica* što se može vidjeti usporedbom Slike 22. i Slike 19.

Skripta omogućuje izradu montaže kartica, ubacivanje XML sadržaja te, ukoliko je potrebno, i numeraciju kartica. U InDesign dokument se pri izvođenju skripte ubacuje XML dokument za prijelom te se preko posebnog izbornika skripte XML elementi povezuju s tekstualnim okvirima. U pojedini tekstualni okvir može se ubaciti tekstualni sadržaj jednog XML elementa ili svih elemenata djece selektiranog XML elementa.

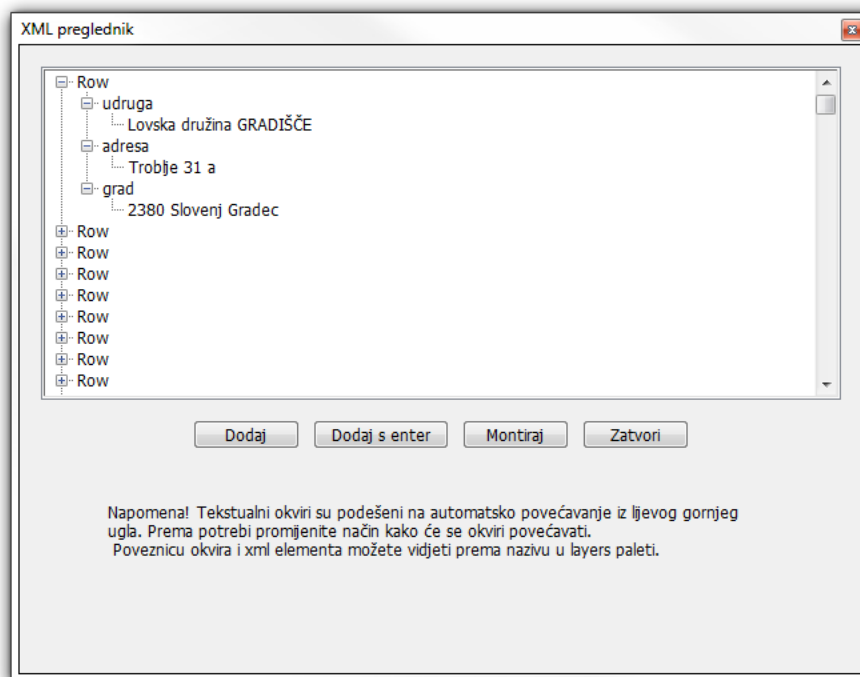


Slika 22. Tijek izvođenja skripte za izradu montaže i personalizacije kartica iz podataka u XML datoteci

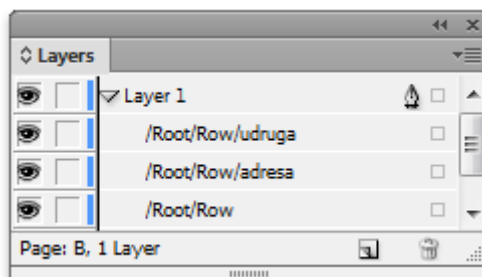
Slike 23. i 24. prikazuju dijaloške okvire koji se kreiraju prilikom izvođenja skripte, a Slika 25. poveznice između XML stabla uvezenog dokumenta i tekstualnih okvira na stranicama dokumenta gdje su veze prikazane u *Layers* paleti programa.



Slika 23. Dijaloški okvir za unos podataka za montažu i broj polja za XML elemente po kartici



Slika 24. XML preglednik: paleta za povezivanje tekstualnih okvira s XML sadržajem



Slika 25. Layers paleta prikazuje poveznice tekstualnih okvira i XML stabla

4.3.6. Skripta za izradu montaže u programu za prijelom (oznaka: S6)

Montaža se u grafičkoj pripremi tipično izrađuje pomoću specijaliziranih programa za izradu montaže. Programi za izradu montaže mogu se podijeliti na 2 skupine: programi koji mogu raditi s bilo kojim sustavom za grafičku pripremu (tipično samostalni programi neovisni o sustavu na kojem rade) i oni koji su uključeni u definirane sustave. U prvu skupinu svrstavaju se programi kao INposition (DK&A), Imposition Publisher (Farrukh Systems), Presswise (ScenicSoft), Strip It (One Vision), Preps (ScenicSoft, Creo, Kodak) i Impostrip (Ultimate). Druga kategorija programa su programi koji se nude od strane poduzeća kao Agfa, Barco, Heidelberg, Scangraphic i Screen u njihovim rješenjima upravljanja proizvodnjom [46].

InDesign ne nudi opcije automatske montaže, osim kod opcija ispisa, a te mogućnosti su ograničene i nedovoljne za svakodnevne poslove u grafičkom poduzeću. Kako je analiziran rad digitalne tiskare, primijećeno je da se značajniji dio vremena troši na izradu montaže jednostavnijih poslova. Često se ovakve pripreme ne izrađuju u programu za montažu, već se izrađuju u programu za prijelom kao što su Adobe InDesign ili Illustrator. Proizvodni proces se ostvaruje na navedeni način upravo zbog toga što je kod jednostavnih poslova proces izrade predloška za montažu ponekad kompleksniji od montaže u navedenim programima.

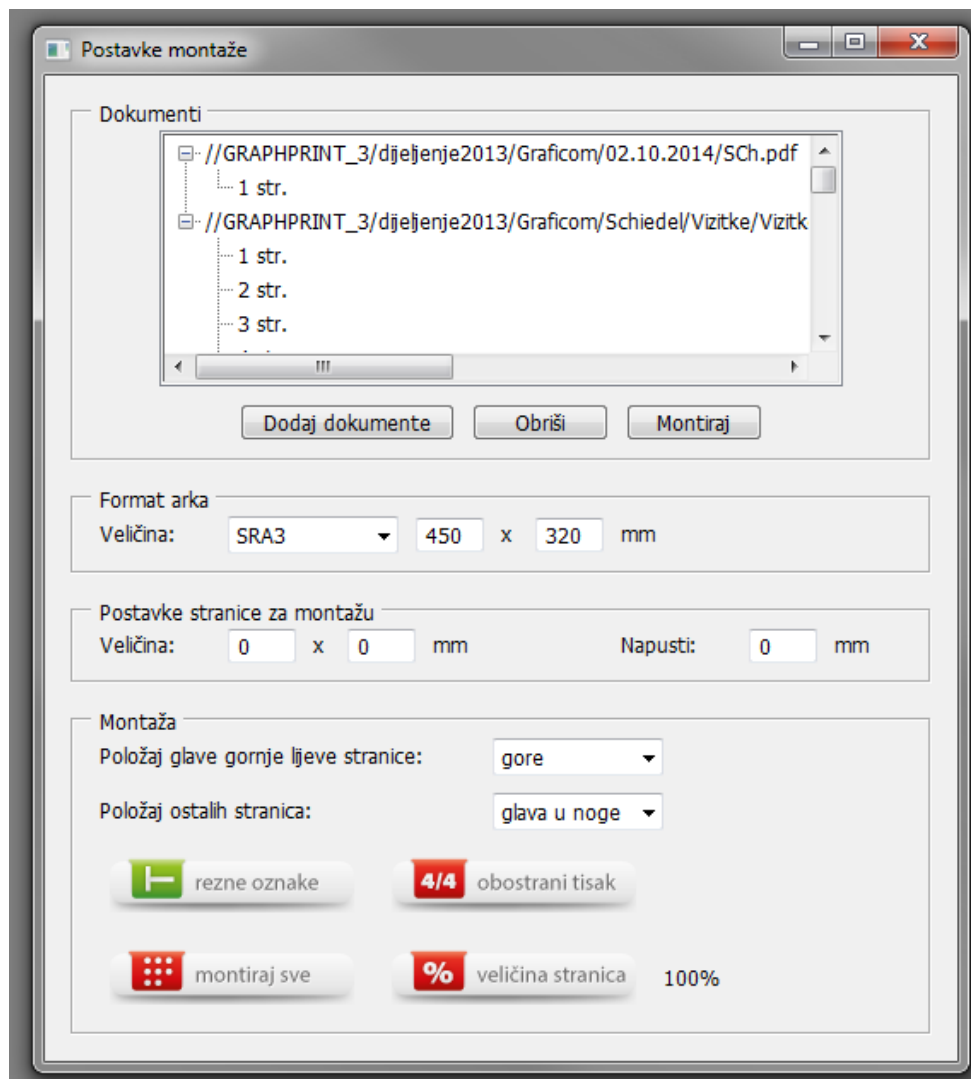
Za složenije se poslove, kao što je montaža knjiga s mnogo stranica za različite vrste uveza, u analiziranom poduzeću koristi program Creo Preps verzije 5.1. U navedenom programu za izradu montaže najprije je potrebno izraditi predložak (*eng. Template*), ukoliko on već ne postoji. Često je za jednostavne poslove potrebno izrađivati nove predloške jer se u njima zadaje kako će se stranice dokumenta smještati po arku, kakvi napusti će se koristiti, veličina stranice, broj stranica po arku, broj ponavljanja pojedinačnih stranica dokumenta... U program se zatim ubacuje jedan ili više PDF dokumenata koji će se montirati, odabire se željeni predložak i ispisuju se montirane stranice dokumenta koje su spremne za tisak.

U slučajevima izrade montaže u programima za prijelom zadaje se veličina tiskovnog arka kao veličina stranice dokumenta na koji se zatim ručno smještaju pojedinačne stranice i njihove rezne oznake u pozicijama na kojima će se tiskati.

Kako se smatralo da je poslove složenijih montaža (npr. montaža knjiga za različite uveze) bolje izrađivati u specijaliziranim programima za prijelom, izrađena je skripta koja izrađuje samo montažu jednostavnijih poslova koji su se ionako izrađivali u navedenim programima. Za poduzeća koja ne posjeduju specijalizirane programe za montažu, mogla bi

se izraditi skripta koja bi radila montažu i za uvez knjiga čime bi se mogli izbjeći troškovi ulaganja u programe za montažu koji su često vrlo skupi.

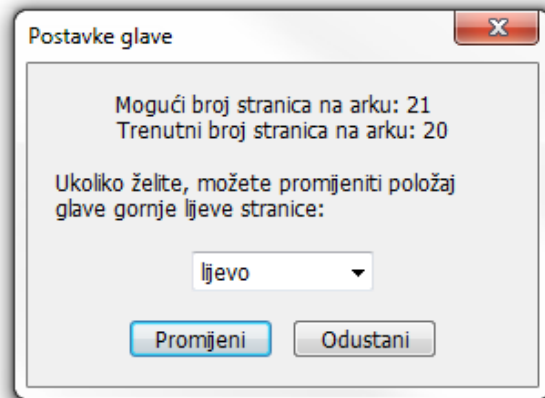
Slika 26. prikazuje dijaloški okvir izrađene skripte. Dijaloški okvir podijeljen je na 4 dijela. Prvi dio *Dokumenti* omogućuje dodavanje dokumenata koji će se montirati. Dodavanje dokumenta pokreće se gumbom *Dodaj dokumente*. Skriptno je zadano dodavanje samo PDF dokumenata. Popis dodanih dokumenata prikazan je u prozoru skripte. Dodani dokumenti reprezentirani su kao stablo koje sadrži putanju do dokumenta ispod koje su smješteni brojevi stranica u dokumentu. Gumb *Obriši* služi za brisanje dodanih dokumenata ili stranica dokumenta koje se ne žele uključiti u montažu. Gumb *Montiraj* služi za pokretanje montaže i pozicioniranje stranica odabranih dokumenata prema postavkama određenim u ostalim dijelovima dijaloškog okvira.



Slika 26. Dialoški okvir skripte za izradu jednostavnih montaža

Drugi i treći dio dijaloškog okvira *Format arka* i *Postavke stranice za montažu* služe za određivanje postavki montaže. U ovim dijelovima izbornika može se selektirati ili unijeti željena veličina tiskovnog arka, veličina stranica i napusta za montažu. Ukoliko korisnik ne zada veličinu stranica za montažu, skripta će iz PDF dokumenta koji je prvi na popisu dokumenta za montažu odrediti koja je veličina stranice i napusta. U tom se slučaju polja za unos veličine stranice i napusta automatski popunjavaju.

Posljednji dio dijaloškog okvira *Montaža* određuje postavke pozicioniranja stranica dokumenta po montažnim arcima. Za montažu se može zadati položaj glave gornje lijeve stranice na prvom arku i zakretanje ostalih stranica. Kada se pokrene montaža, a navedene postavke su postavljene tako da na arak stane manje stranica (manja iskoristivost montažnog arka), skripta će izraditi novi dijaloški okvir gdje će pitati korisnika želi li izmijeniti postavke pozicioniranja stranica tako da više stranica stane na arak (Slika 27.).



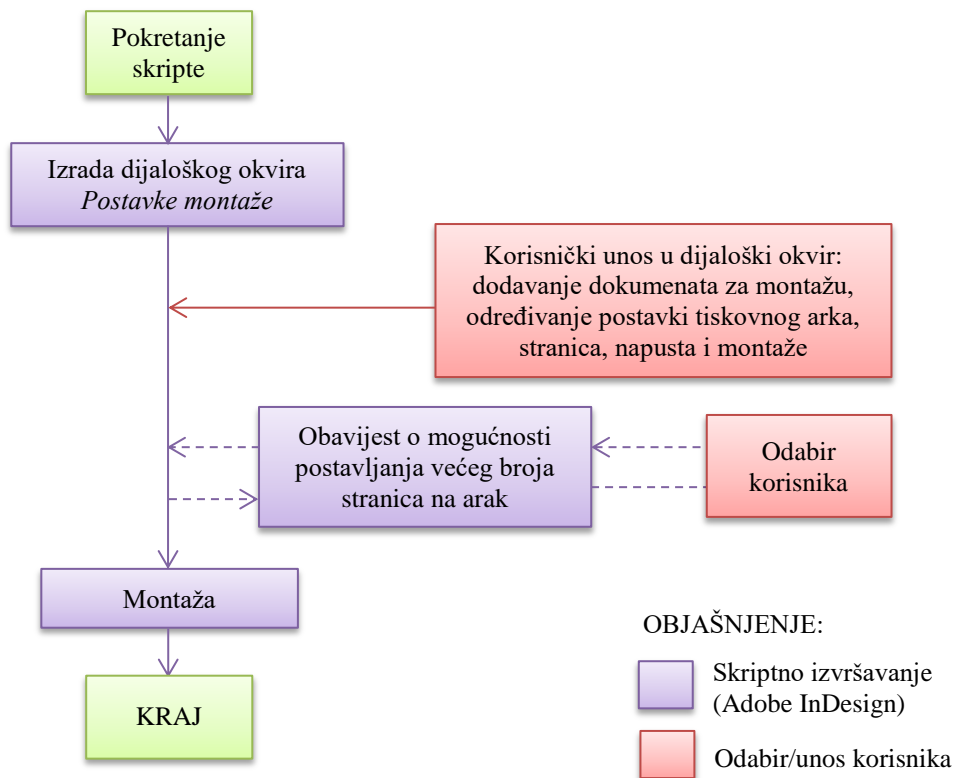
Slika 27. Dialoški okvir koji se prikazuje u slučajevima postavki montaže gdje se postavlja manje od mogućeg broja stranica na selektiranoj veličini tiskovnog arka

Gumb *rezne oznake* omogućuje uključivanje i isključivanje crtanja reznih oznaka. Opcija *obostrani tisak* omogućuje smještanje stranica na arku za obostrani tisak. Uključenjem navedene opcije, ovisno o montaži, mijenja se raspored stranica na arku i one se prema potrebi zakreću. Opcija *montiraj sve* omogućuje montažu u kojoj se više različitih stranica smješta na isti arak. Bez uključivanja ove opcije na jednom se arku nalazi montirana samo jedna stranica dokumenta. Posljednja opcija *veličina stranica* omogućuje unos postotka skaliranja svih stranica PDF dokumenta prije montaže. Navedena opcija utječe samo na stranice PDF dokumenta koji se ubacuje, a ne utječe na postavke veličine stranice za montažu koje su zadane u trećem dijelu dijaloškog okvira.

Slika 28. prikazuje gotovu montažu izrađenu u programu InDesign uz pomoć skripte, a Slika 29. tijek izvršavanja skripte.



Slika 28. Montaža izrađena pomoću skripte u programu InDesign



Slika 29. Tijek izvođenja skripte za montažu

4.3.7. Skripta za automatizaciju izrade ePub dokumenta koji sadrži SVG datoteke (oznaka: S7)

Grafičko izdavaštvo suočava se s rastom potražnje izrade digitalnih publikacija. Izumom tableta, odnosno e-čitača, stvoren je novi medij za izdavaštvo knjiga. Posebnost digitalnih knjiga svakako je u mogućnostima interaktivnosti sa sadržajem koji pružaju. Ta činjenica ne odnosi se samo na tekst, već i na sve ostale elemente multimedije kao što su slike, video, zvuk i animacija. Interaktivne knjige široko su rasprostranjene te su revolucionirale način na koji čitatelji konzumiraju sadržaj. S razvojem softvera povećala se mogućnost jednostavnog kreiranja interaktivnih e-knjiga, a s obzirom na potražnju na tržištu povećali su se zahtjevi za kvalitetom i standardi samog značenja riječi interaktivnost. Kreiranje nove e-knjige ili konvertiranje tradicionalno otisnutih knjiga u digitalni oblik predstavlja ogroman potencijal na tržištu s obzirom na eksponencijalan rast potražnje posljednjih nekoliko godina. Prodaja digitalnih knjiga sve više nadmašuje prodaju tiskanih knjiga. Razlog tomu je činjenica da su e-knjige jeftinije od tiskanih, ne zauzimaju mnogo prostora i olakšana je dostava [47].

Analize tržišta ukazuju na porast korisnika e-knjiga kako raste broj prodanih specijaliziranih čitača. E-knjige su često prvi odabir kupaca, iako tiskana izdanja i dalje prevladavaju [48]. Čitači za čitanje e-knjiga i formati digitalnih izdanja se razlikuju. Neke formate nije moguće čitati bez specijalnih uređaja i programa namijenjenih za njihovo čitanje kao npr. Kindle uređaj i njegov AZW format dostupan na Amazon web stranicama. Ipak, posljednjih nekoliko godina mnogi uređaji i aplikacije počinju prihvaćati širi raspon formata, primjerice PDF, običan tekst i ePub izdanja.

EPub (eng. *electronic publishing format*) je otvoreni standard kojega je razvio IPDF (*International Digital Publishers Forum*) [49]. Postao je standard 2007. godine, a trenutna verzija je ePub 3.0.1. ePub je baziran na web standardima i daje načine reprezentiranja, pakiranja i kodiranja strukturiranog i semantičkog web sadržaja koji uključuje XHTML, CSS, SVG, slike i ostale resurse koji se dijele u jednoj datoteci. Omogućuje izdavačima da proizvedu digitalnu publikaciju. ePub 2 je bio inicijalno standardiziran 2007. godine kao nasljednik „*Open eBook Publication Structure*“ ("OEB") koji je početno standardiziran 1999. godine. ePub 2.0.1 odobren je 2010. godine i bio je posljednje ePub 2 izdanje. Godine 2011. ePub 3 zamjenjuje ePub 2 kad je odobren u posljednjoj fazi (*Recommended Specification*), a 2014. godine bila je posljednja nadogradnja na ePub 3.0.1 te je navedena verzija ujedno i posljednja verzija ePub dokumenta [50].

Datoteke koje su izrađene kao EPUB dokumenti imaju ekstenziju *.epub*. EPub je izrađen kao dokument čiji se sadržaj prilagođava za pojedine uređaje, iako može biti izrađen i kao dokument s fiksnim prijelomom.

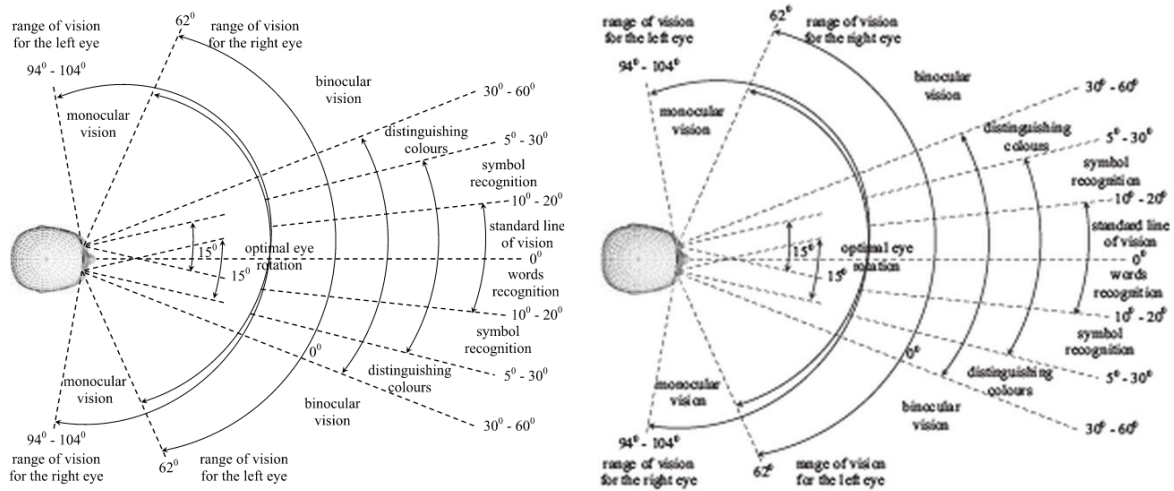
EPub format prihvaćen je od većine prodavača digitalnih izdanja uključujući Google, Apple's iBookstore, Barnes and Noble's Nook i Kobo. Amazon ne prihvaća navedeni format, ali će ga prihvatiti izravno od izdavača i prebaciti ga u vlastito Kindle izdanje. Iz navedenih razloga ePub je primarni format za mnoge izdavače, iako se većina njihove prodaje obavlja na Amazonu [51].

Uobičajeno se prilikom izrade ePub izdanja izrađuje samo jedan ePub dokument za pojedinu knjigu koji se zatim predaje na različite internetske stranice koje prodaju digitalna izdanja. Ta verzija mora se dobro prikazivati kroz različite čitače e-knjiga koji imaju različitu veličinu ekrana, rezoluciju, podršku za boju i programe za renderiranje. Jedan je od faktora koji utječe na izradu ePub dokumenta da neki prodavači digitalnih izdanja (npr. Amazon) zaračunavaju trošak preuzimanja datoteka prema njihovoj veličini. Ukoliko se u ePub izdanju nalaze mnoge velike rasterske slike, takav trošak može biti nepotrebno velik. Kako se ePub dokumenti često izrađuju iz dokumenta koji su pripremani za tisak, može se javiti i dodatni problem, tj. da neke rasterske slike nisu podobne za objavu u digitalnim izdanjima. Većinom su to slike nastale iz vektorskih datoteka koje sadrže tekstualne podatke gdje tekst pretvorbom u raster postaje nečitljiv. Kako bi se izbjegle pretvorbe u nečitljive rasterske datoteke, moguće je koristiti SVG format datoteka za objavu u ePub dokumentima. Danas je SVG sveprisutan u modernim web preglednicima i sve češće se koristi za prikaz vektorskih podataka na webu. Slika 30. prikazuje usporedbu dobivenih slika u *Chrome* pregledniku kada je vektorska slika pretvorena u raster i kada se ona objavljuje kao SVG datoteka. Manje je poznato da je SVG podržan i u programima za čitanje e-knjiga te je dio ePub standarda [52].

U analiziranom grafičkom poduzeću, uz tisak manjih naklada knjiga, klijenti često zahtijevaju izradu digitalnih izdanja. Iz programa InDesign vrlo se lako izvoze ePub dokumenti. Glavni nedostatak izrade ePub dokumenta iz programa Adobe InDesign je kvaliteta izvezenih vektorskih crteža. U programu ne postoji mogućnost automatske izrade ePub dokumenta koji sadrže SVG datoteke pa je tako mogućnost digitalnih izdanja s vektorskim crtežima ograničena.

U analiziranom poduzeću poslovi dodavanja SVG-a u ePub dokument izrađivani su naknadno, nakon izvoženja dokumenta. Standardno odvijanje zadatka izrade ePub dokumenta sa SVG datotekama bilo bi kako slijedi:

U ePub paket ubacuje se SVG verzija crteža i u XHTML datoteci ePub dokumenta mijenja se kod koji povezuje crtež. SVG se izrađuje iz vektorskih crteža u *.ai*¹⁶ ili *.eps*¹⁷ datotekama izvoženjem iz programa Adobe Illustrator.



Slika 30. Pregled snimke ekrana vektorskog crteža s tekстом objavljenim kao SVG (lijevo) i PNG datoteka (desno) u Chrome web pregledniku

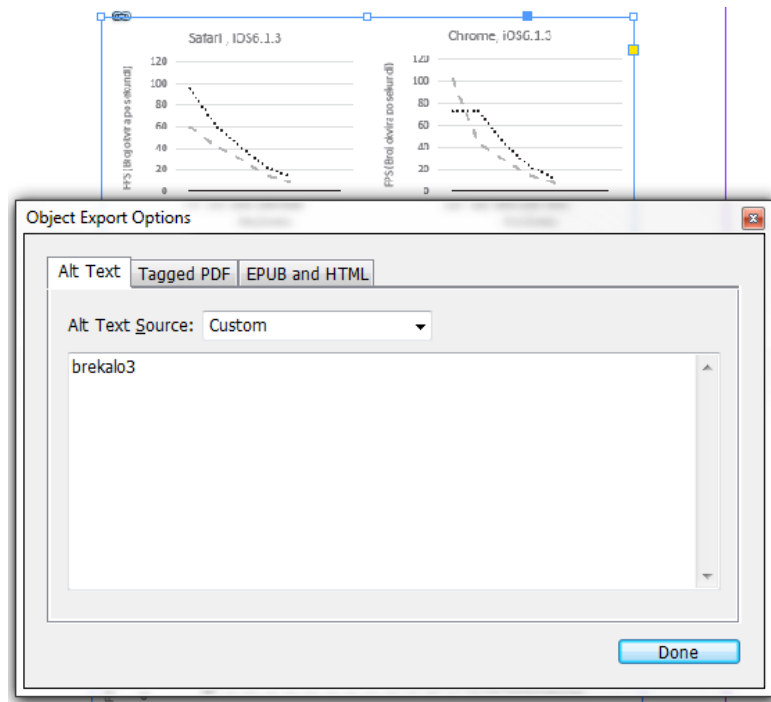
Zadatak skripte je omogućiti automatsku izradu i optimizaciju izvoženja ePub dokumenta koji sadrži SVG vektorske grafike. Izrađena automatizacija uključuje 2 skripte. Prva skripta se provodi prije izrade ePub dokumenta iz Adobe InDesign programa. Zadaća navedene skripte je naći sve poveznice s *.ai* ili *.eps* datotekama te iz navedenih datoteka izraditi *.svg* datoteke. Skripta pokreće program Adobe Illustrator te u njemu otvara originalnu datoteku, postavlja potrebne postavke na crtež i sprema ga kao *.svg* format u poseban direktorij naziva „*SVG_Folder*“. Navedeni direktorij kreira se na istom mjestu u računalu gdje je spremljena i originalna *.indd* datoteka. Nakon izvršenja prve skripte izvozi se ePub dokument. U nastalom ePub dokumentu na mjestima poveznica na vektorske datoteke postavljene su *.png* datoteke čija se imena razlikuju od originalnih naziva vektorskih datoteka iz kojih su izrađene. Kako bi se poveznice mogle u drugoj skripti izmijeniti na način da se na mjestu *.png* datoteka postave odgovarajuće poveznice na *.svg* datoteke, prva skripta postavlja preko „*Object Export Options*“ naziv *.svg* datoteke za zamjenu pod „*Alt Text*“ opcijom za izvoženje (Slika 31.). Na Slici 32. prikazan je tijek izvođenja prve skripte.

Nakon izvršenja prve skripte potrebno je izraditi ePub dokument i otpakirati ga upotrebom nekog od programa za otvaranje *.zip* datoteka. Nakon što su datoteke otpakirane,

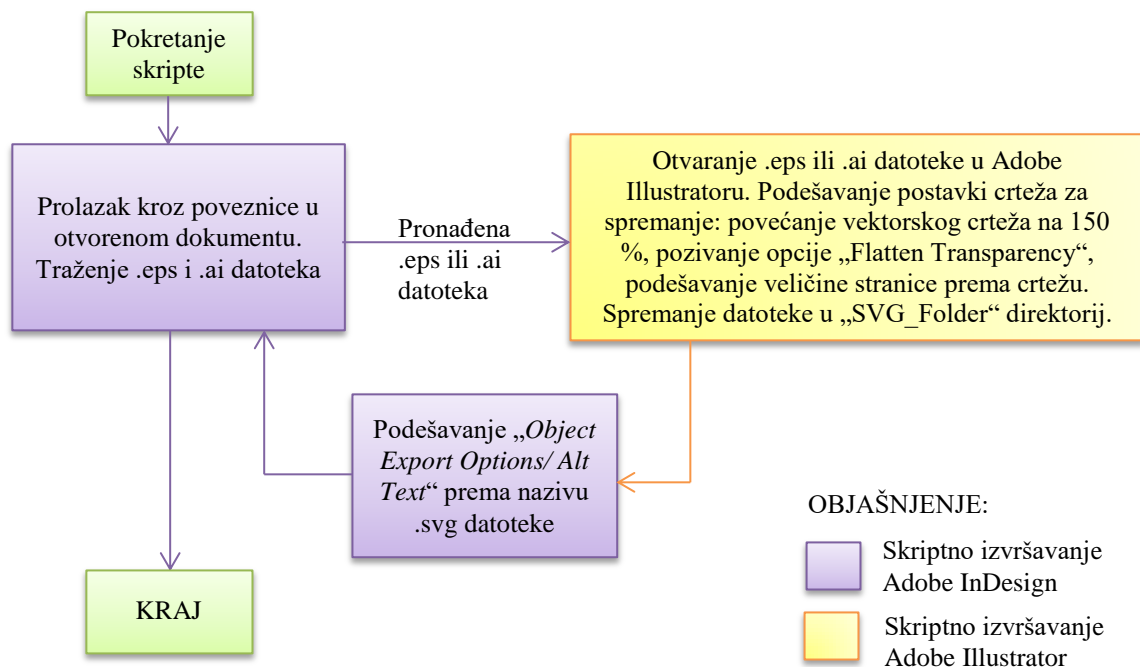
¹⁶ Ekstenzija Adobe Illustrator vektorske datoteke

¹⁷ Vektorska datoteka u formatu EPS (eng. *Encapsulated postscript*)

pokreće se druga skripta koja briše *.png* datoteke, mijenja *.html* dokument ePub publikacije (Slika 33.) te ubacuje *.svg* datoteke u odgovarajući direktorij ePub dokumenta.



Slika 31. Postavljena „Object Export Options/ Alt Text“ opcija preko skripte koja sadrži naziv *.svg* datoteke koja će se povezati preko druge skripte



Slika 32. Tijek izvođenja prve skripte za izradu ePub dokumenta koja uključuje izradu *.svg* datoteka i podešavanje *.indd* dokumenta za izvoženje

Nakon što je skripta izvršila zadatke izmjene, potrebno je ponovno zapakirati dokument kao *.epub*. Na Slici 34. prikazan je tijek izvođenja druge skripte za izradu ePub dokumenta koji sadrže *.svg* datoteke.

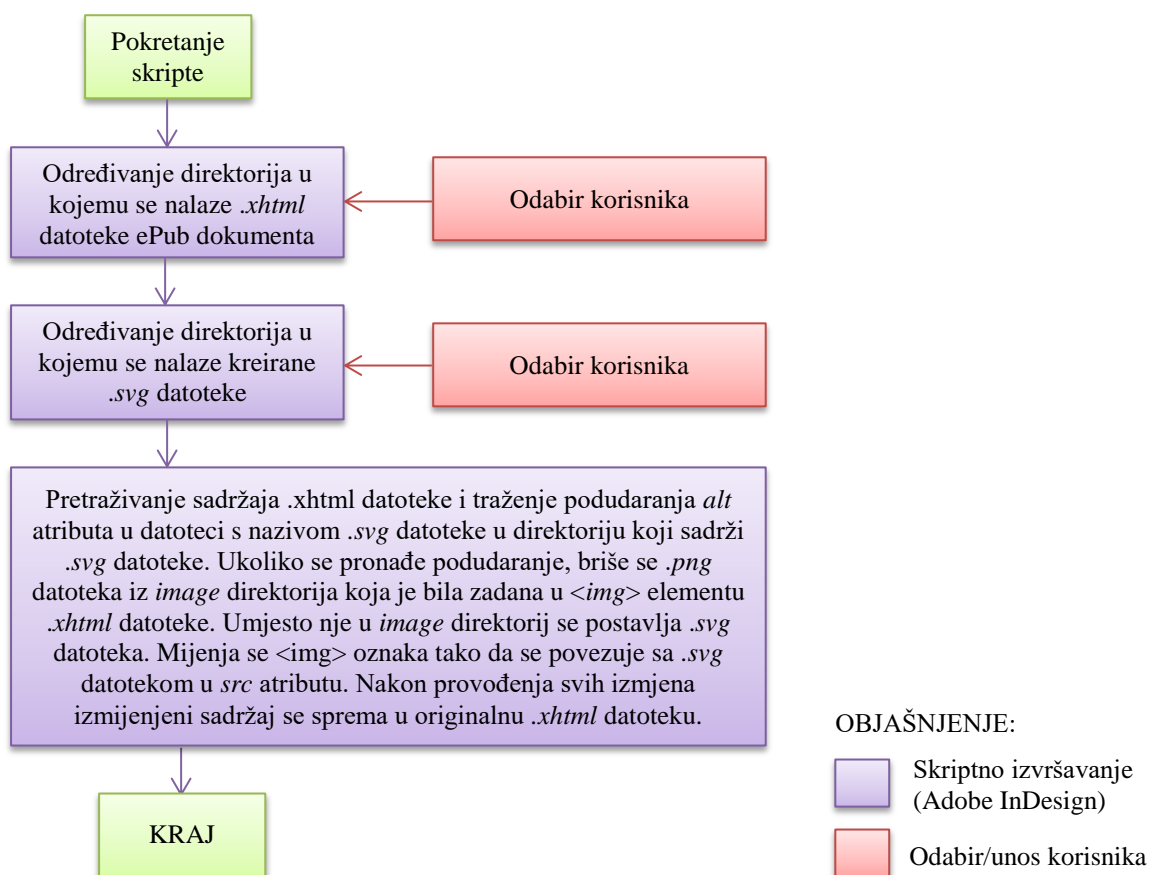
```
<div class="frame-28">
  
</div>
```

a) izgled dijela koda *.xhtml* datoteke prije promjene

```
<div class="frame-28">
  
</div>
```

b) izgled dijela koda *.xhtml* datoteke nakon promjene

Slika 33. Dio koda *.xhtml* datoteke prije (a) i nakon (b) izvođenja druge skripte



Slika 34. Tijek izvođenja druge skripte za izradu ePub dokumenta

4.3.8. Skripta za automatsku obradu fotografija (oznaka: S8)

Pri izradi pripreme za tisak veliki se dio vremena troši na obradu i pripremu fotografija za tisak. Fotografije koje se koriste u prijelomu, ovisno o tehnici tiska, moraju zadovoljavati određene standarde. Ukoliko ne zadovolje barem minimalne standarde za reprodukciju, kvaliteta tiska bit će uvelike smanjena.

Standardi i preporuke pri obradi fotografija za tisak navode sljedeće:

- Rezolucija slika za većinu tiskarskih tehnika mora biti 300 dpi. Rezolucija manja od 260 dpi značajno smanjuje kvalitetu slike na otisku.
- Kako bi se slike mogle otisnuti u očekivanim bojama, moraju biti pretvorene u CMYK prostor boja te obrađene. Prebacivanje u CMYK prostor radi se prije obrade slike. U slučaju da slike ostanu u RGB prostoru i budu tiskane, doći će do velike i nepravilne transformacije boja.
- Ukoliko nije drugačije zadano, preporučena metoda renderiranja je relativno kolorimetrijsko renderiranje.
- Na mnogim tiskarskim strojevima slike izgledaju tamnije nakon tiska zbog pozitivnog prirasta raster tonskih vrijednosti u tisku. Zbog navedenog razloga slike je prilikom obrade poželjno posvijetliti. Povećavanje svjetline radi se za otprilike 10 % ili više, ovisno o motivu i kvaliteti slike.
- Kod nekih tehnika tiska radi se korekcija boja za tisak. Ofsetni tisak koristi tekuće boje koje su upojne. Naime, kod manje kvalitetnih i upojnih papira boja toliko uđe u papir da na površini izgubi intenzitet pa ju je potrebno pojačati.
- Kada se slika gleda na monitoru, ljudsko oko je doživljava oštrije nego na tiskovnoj podlozi zbog jednakog pozadinskog osvjetljenja na cijeloj slici. Zbog toga je navedenu oštrinu za tisak potrebno umjetno povećati. Ovaj problem izrazito je vidljiv na slikama koje imaju fine detalje. Izoštavanje slike jedan je od najvažnijih zadataka koji se moraju obaviti prije tiska. Ukoliko se slika koja se uređuje ispisuje na rezoluciji 300 piksela po inču, rubovi slike trebaju radijus od 3 piksela kako bi izgledali oštri u tisku [53].

Osim navedenih parametara obrade slika, u različitim računalnim programima slike se mogu dorađivati i alatima kojima se kvaliteta slike može još povećati, ali to je stvar procjene i sposobnosti onoga koji slike obrađuje [54].

Gotovo svaka slika dobivena od strane naručioca prije tiska prolazi kroz nabrojena uređenja. Većina materijala dostavljenih za pripremu za tisak dolaze kao Word dokumenti s umetnutim slikama. Ponekad se dostavljaju i slike kao odvojene datoteke, ali gotovo uvijek neobrađene i nespremne za tisak. Kako navedeno uređivanje oduzima dosta vremena pri pripremi dokumenata za tisak, proces uređivanja i obrade slika želio se automatizirati.

Automatizacija obrade slika za tisak može se izraditi do određenog stupnja jer zaseban pristup pojedinoj datoteci može dati bolje rezultate nego automatizirane skripte koje obrađuju slike jednakim pristupom za sve slikovne datoteke, neovisno o njihovom izvoru. Obrada slika, ovisno o upotrijebljenom sustavu, iskustvu i sposobnostima osobe koja obrađuje slike, može dati mnogo bolje rezultate nego uniformna obrada svih slika u publikaciji. Često je vrlo teško i vremenski zahtjevno bar minimalno uređenje slika za pripremu za tisak prema ranije nabrojenim preporukama.

Kako bi se ubrzala digitalna grafička priprema, željelo se na slike postaviti barem minimalno uređivanje bez obzira na izvor i kvalitetu slike. Prema potrebi, slike se tada kasnije mogu dodatno urediti.

U analiziranom se grafičkom poduzeću prijelom publikacija koje sadrže fotografije izrađuje tako da se pri izradi prijeloma najprije koriste originalne slike, koje se nakon odobrenog prijeloma pripremaju za tisak, ovisno o tehnici tiska. Nakon odobrenog prijeloma slike se pojedinačno otvaraju u programu Adobe Photoshop i uređuju prema ranije nabrojenim standardima i preporukama. Nakon probnog otiska na stroju za digitalni tisak uz korištenje *Color Management* sustava slike se ponekad moraju dodatno podesiti kako bi otisak bio zadovoljavajući.

U program za prijelom obrađene se slike ubacuju u *.psd* formatu kako bi se prema potrebi mogle naknadno urediti bez gubitka kvalitete koji su prisutni u ostalim komprimiranim formatima rastiranih datoteka.

Zadatak skripte za automatsku obradu fotografija je unaprjeđenje procesa obrade fotografija za tisak. Nakon izrađenog prijeloma pokreće se skripta koja uređuje slike kako bi ih pripremila za tisak. Skripta prolazi kroz sve poveznice na *.jpg*, *.jpeg*, *.tif*, *.tiff* i *.png* datoteke u dokumentu te uređuje slike sljedećim postavkama u Adobe Photoshop programu:

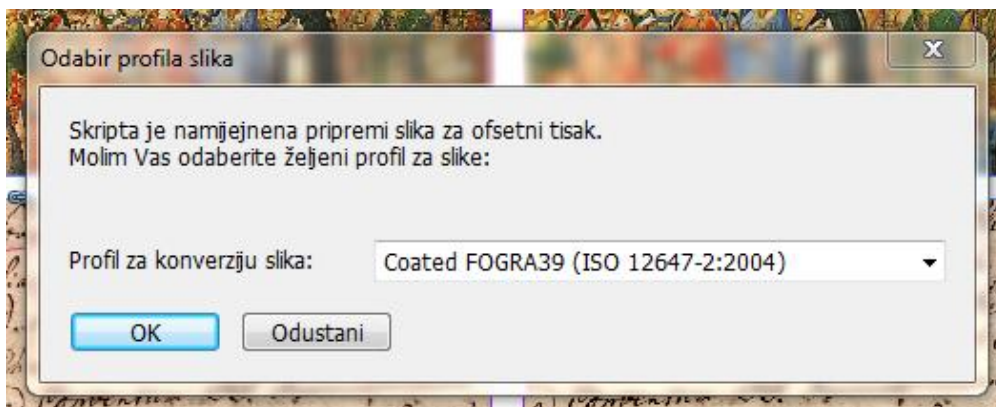
- Ukoliko je potrebno, slika se proporcionalno povećavana *effective ppi* u *InDesignu* na otprilike 300 dpi u koracima od 110 % širine i visine slike s *Resample Method* postavljenom na *Bicubic Smoother*. Ovakav način preporučeno se primjenjuje pri povećavanju slike koja nije zadovoljavajuće rezolucije [55]. Uloga je grafičkog dizajnera da već pri izradi prijeloma

odbacuje fotografije koje su premale kvalitete da bi se mogle zadovoljavajuće reproducirati.

- Postavljenje zadanog CMYK profila boja (Coated FOGRA39 (ISO 12647-2:2004) ili 'Uncoated FOGRA99 (ISO 12647-2:2004)) s *Relative Colorimetric Intent*.
- Podešavanje krivulja slike dodavanjem *Curves* sloja te snižavanje vrijednosti srednjih tonova slike kako bi se u tisku osigurao veći raspon boja zbog prirasta tonova u slici.
- Osnovno podešavanje *Shadows/Highlights* opcije slike.
- Podešavanje i primjenjivanje *Reduce Noise* filtra [56].
- Kopiranje *Background* sloja, postavljanje *Blending Mode* na *Overlay* i postavljanje *High Pass* filtra za izoštravanje slike [56].

Nakon što je slika obrađena, sprema se kao *.psd* datoteka. U programu Adobe InDesign potrebno je izmijeniti poveznicu na novonastalu datoteku. Slike se spremaju kao *.psd* datoteke kako bi se kasnije, nakon izrade probnog otiska, mogle prema potrebi dodatno urediti.

Slika 35. prikazuje dijaloški okvir skripte za selekciju željenog profila koji će se primijeniti na slike. Slika 36. prikazuje kako su originalne slike izmijenjene i pripremljene za tisak pomoću skripte. Pri ispisu ovog rada, pri izradi otiska Slike 36. korištena je originalna PDF datoteka izrađena iz Adobe InDesign programa kako bi se mogle bolje vidjeti dobivene razlike u fotografijama. Navedeni dokument prilog je elektronskom izdanju rada. Tijek izvođenja skripte prikazan je na Slici 37.



Slika 35. Dijaloški okvir odabira profila boja za primjenu na fotografije

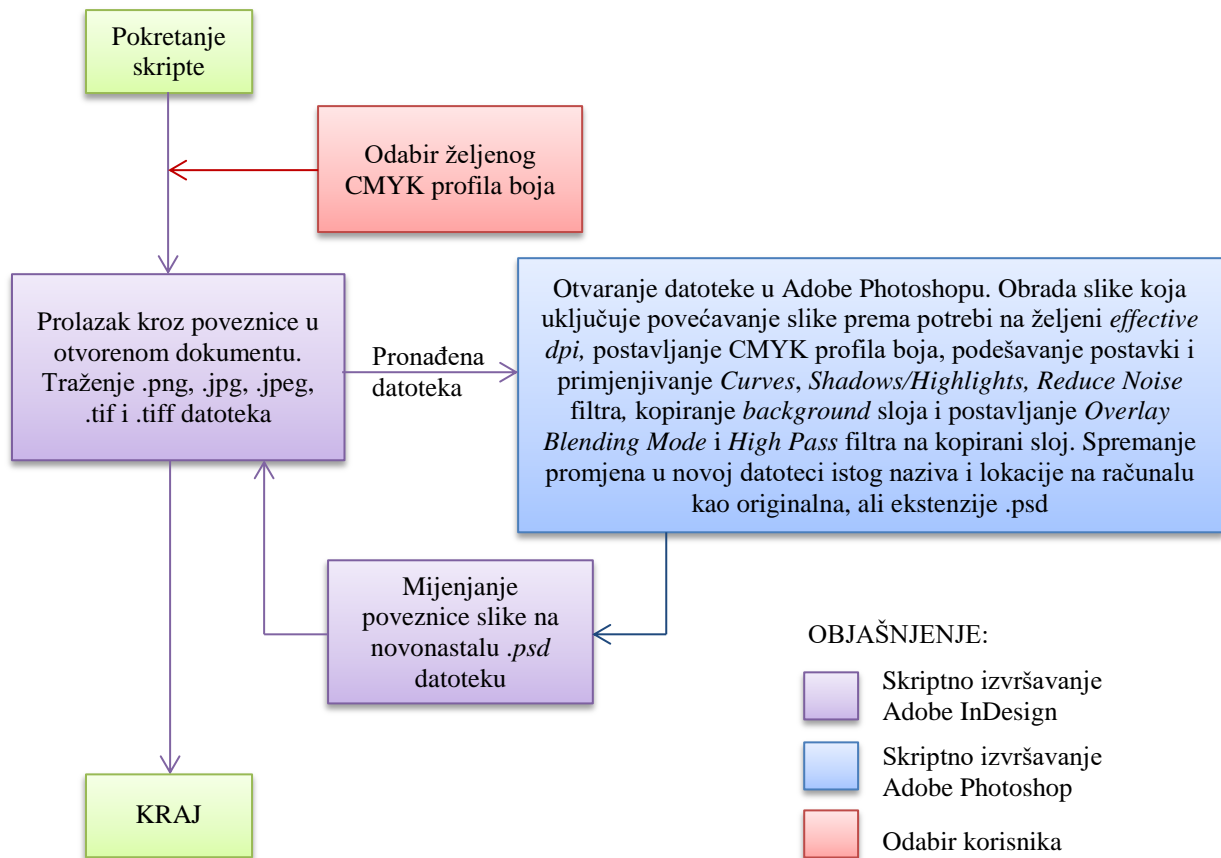


Novembrio Auctoritate Regia per Illm
liamum Vrbhovan Epum Zagrabiensem,
D. Franciscus Raffaj et Paulo Selt
Charmensis Canonius, Barochia S. M
ad Conventum S. S. Franciscanorum
in La Barochus institutus, et Canonica
tilarion Lavaprich.

Novembrio Auctoritate Regia per Illm
liamum Vrbhovan Epum Zagrabiensem,
D. Franciscus Raffaj et Paulo Selt
Charmensis Canonius, Barochia S. M
ad Conventum S. S. Franciscanorum
in La Barochus institutus, et Canonica
tilarion Lavaprich.



Slika 36. Prikaz obrade slika skriptom. Lijeva strana prikazuje originalnu datoteku, desna strana sliku obrađenu pomoću skripte
(Izvor slika za obradu: Časopis Zvona, List Čakovečkih Župa, br. 10, 2014.)



Slika 37. Tijek izvođenja skripte za obradu fotografija

4.3.9. Skripta koja uključuje izradu barkoda iz varijabilnih podataka (oznaka: S9)

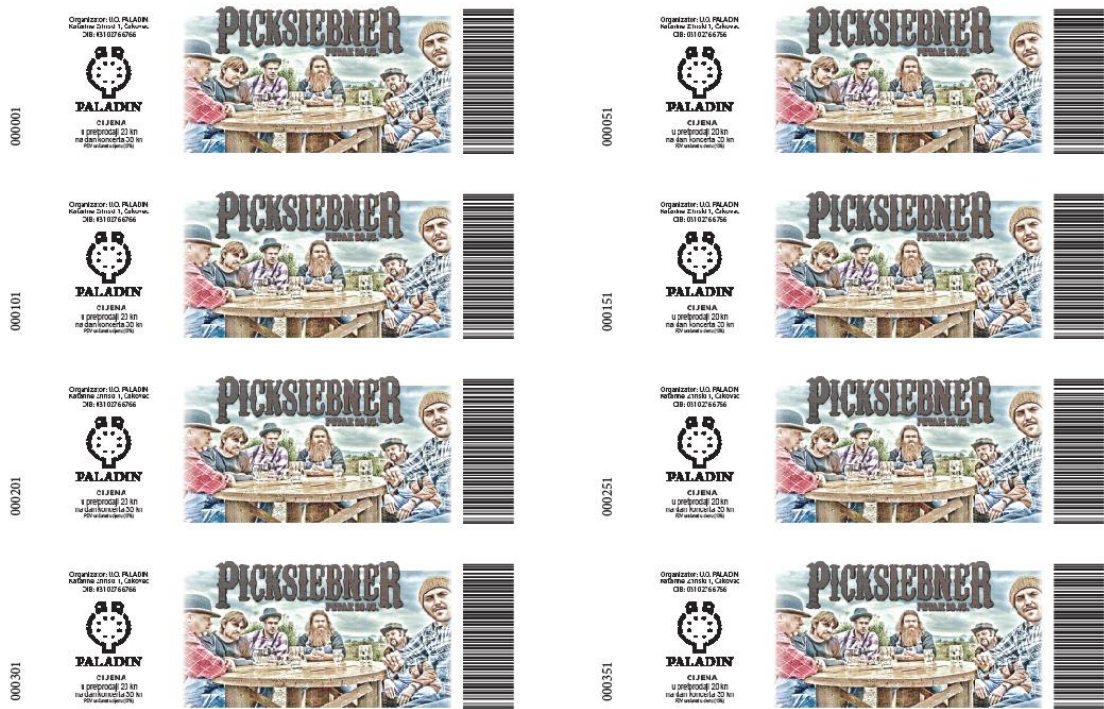
Izrada varijabilne grafičke pripreme koja uključuje izradu promjenjivog barkoda optimalno se može izrađivati u specijalnim komercijalnim alatima koji su često ograničeni na specijalne vrste digitalnih strojeva i zahtijevaju znatna ulaganja u alate rada. Bez upotrebe specijalnih alata za grafičku pripremu posao izrade navedene pripreme u standardnim alatima za grafičku pripremu je dugotrajan, često neisplativ i sklon pogreškama. Budući da program Adobe InDesign ne omogućuje izradu barkoda, potrebno je koristiti druge, često komercijalne alate za izradu barkodova. Bez upotrebe navedenih komercijalnih alata izrada većeg niza kartica s barkodom bila bi izuzetno dugotrajna (npr. upotrebom besplatnih servisa na web stranicama) jer bi svaki barkod trebalo izraditi pojedinačno i nakon toga ubacivati u prijelom. Nedostatak upotrebe komercijalnih alata za VDP je potreba kupnje alata za izradu barkodova.

Izgled konačne pripreme i željeni rezultat skripte prikazani su na Slici 38. Za izradu pripreme korištene su 2 skripte. Prva je skripta za izradu montaže (i numeracije prema potrebi) te za pozicioniranje i povezivanje tekstualnih okvira za barkod. Druga je skripta koja omogućuje izradu Code 128 barkoda iz cjelokupnog sadržaja tekstualnog okvira, odabranog dijela teksta ili tablice s alfanumeričkim zapisom sadržaja koji će se kodirati. Druga skripta izrađena je odvojeno od numeracije i montaže kako bi se mogla upotrebljavati i za druge poslove koji ne zahtijevaju navedene zadatke. Time se želio zadovoljiti kriterij pisanja skripti koje se mogu upotrebljavati u različitim proizvodnim procesima.

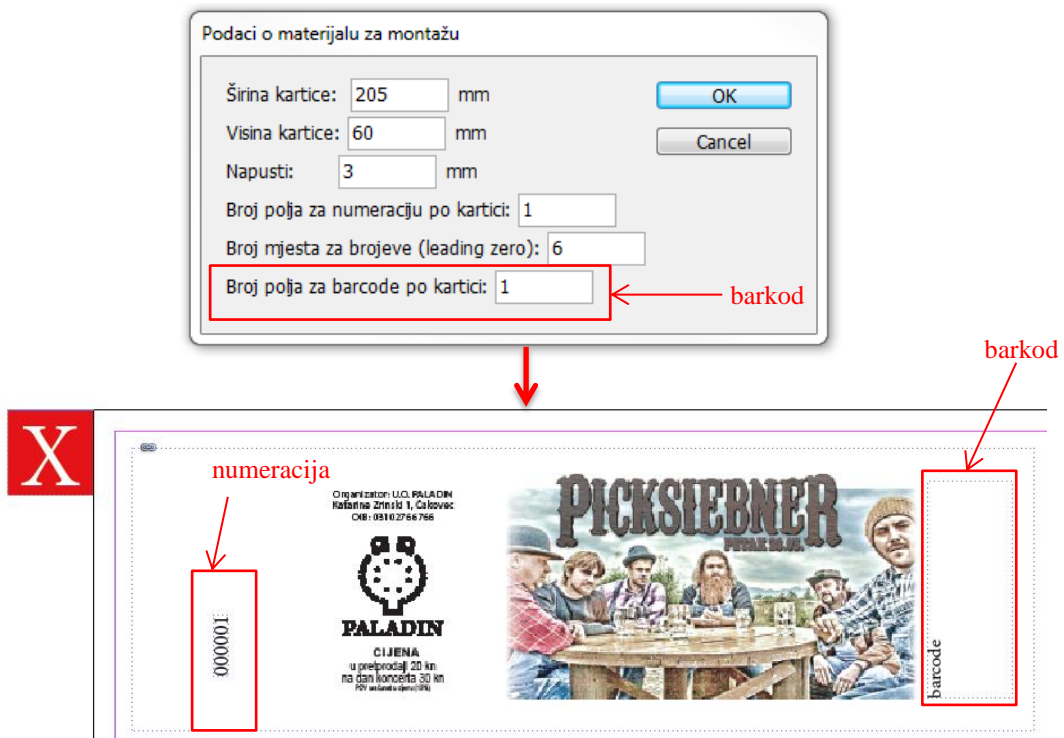
Cijeli proces prve skripte odvija se kako je opisano u 4.3.4. *Skripta za izradu numeracije i montaže kartica*. Jedina je razlika u tome što su polja za numeraciju opcionalna te je dodana mogućnost ubacivanja polja za barkod (Slika 39. i 43.). Polja za barkod nemaju skriptno ubačen sadržaj, to su tekstualni okviri koji se međusobno povezuju ovisno o odabranim postavkama numeracije (Slika 40.).

Druga skripta ovisi o korisnikovoj selekciji. Ukoliko je selektiran tekstualni okvir ili raspon odlomaka u priči¹⁸, izradit će se više kodova odjednom. Skripta omogućuje izradu svih verzija *Code 128* barkoda što je vidljivo na Slici 42. Slika 44. prikazuje tijek izvođenja druge skripte za izradu *Code 128* barkoda.

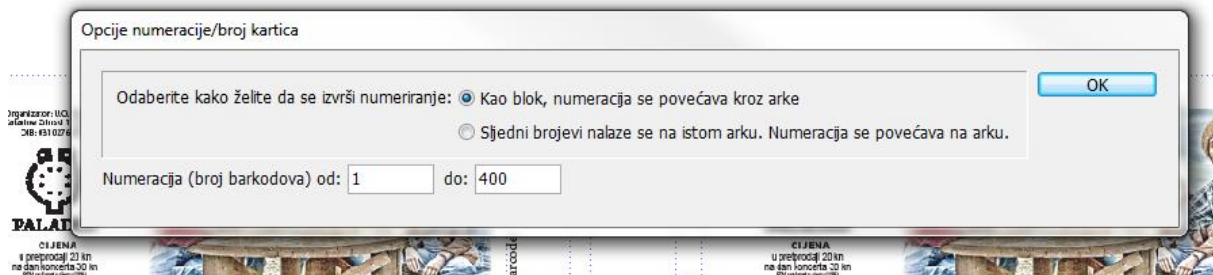
¹⁸ eng. *Story* označava povezane (eng. *Linked*) tekstualne okvire u Adobe InDesign programu



Slika 38. Montaža, numeracija i postavljanje Code 128 barkoda, konačna priprema spremna za tisak



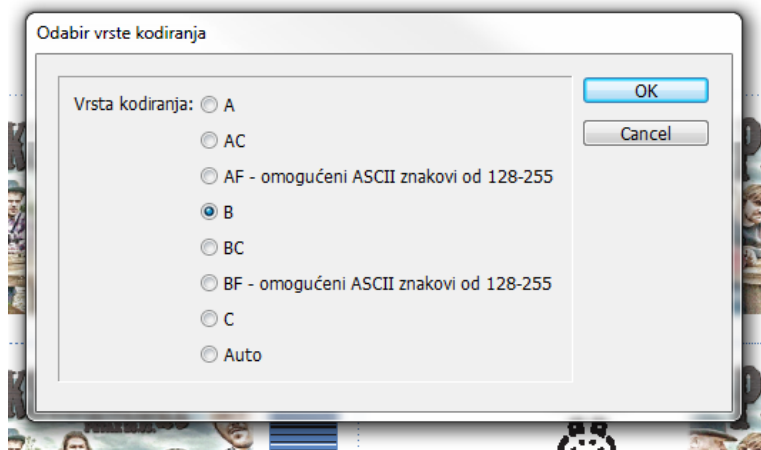
Slika 39. Prikaz polja za barkod u skripti za numeraciju i montažu



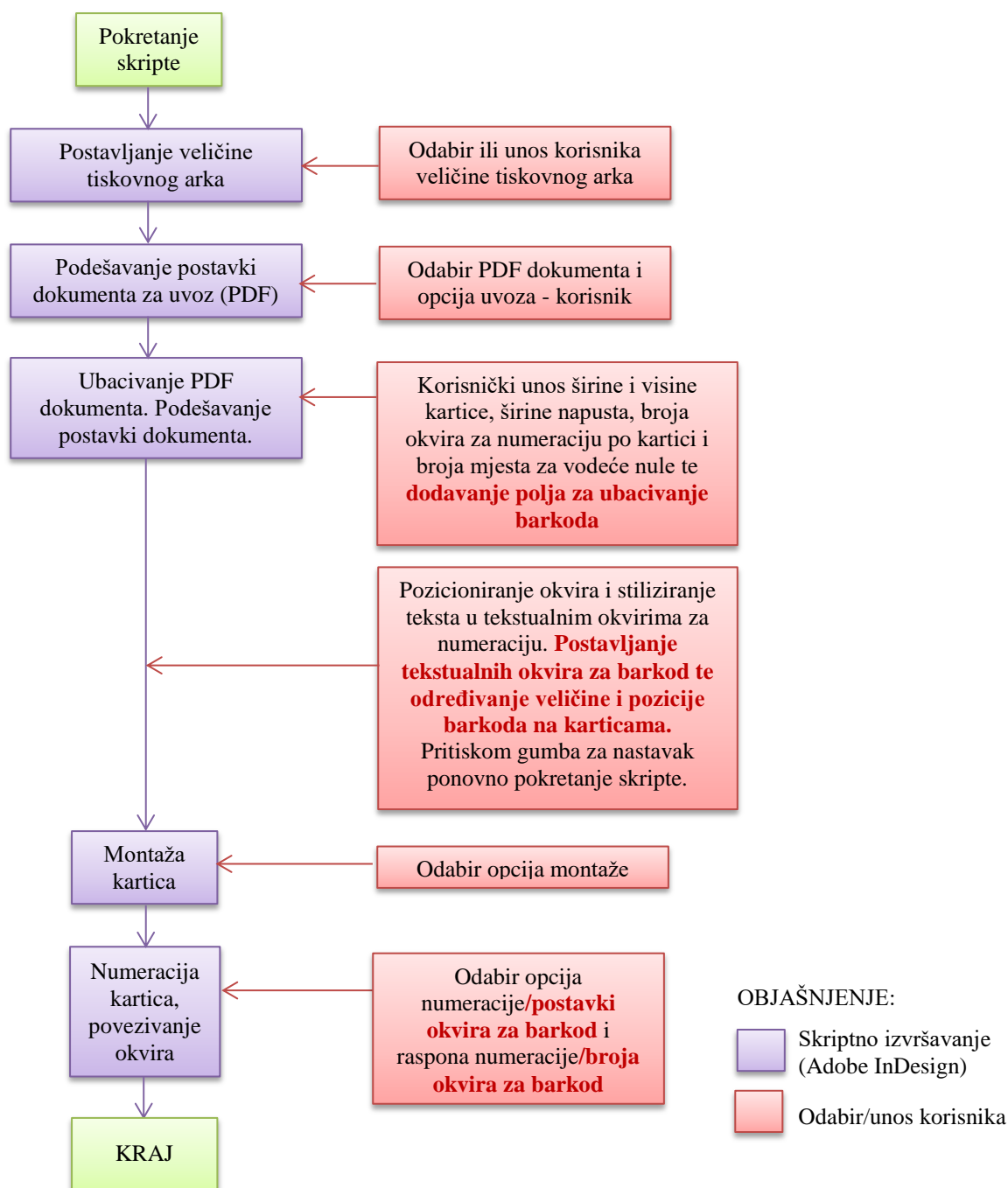
Slika 40. Prikaz upisivanja raspona numeracije i/ili broja barkodova koji će se dodati u dokument



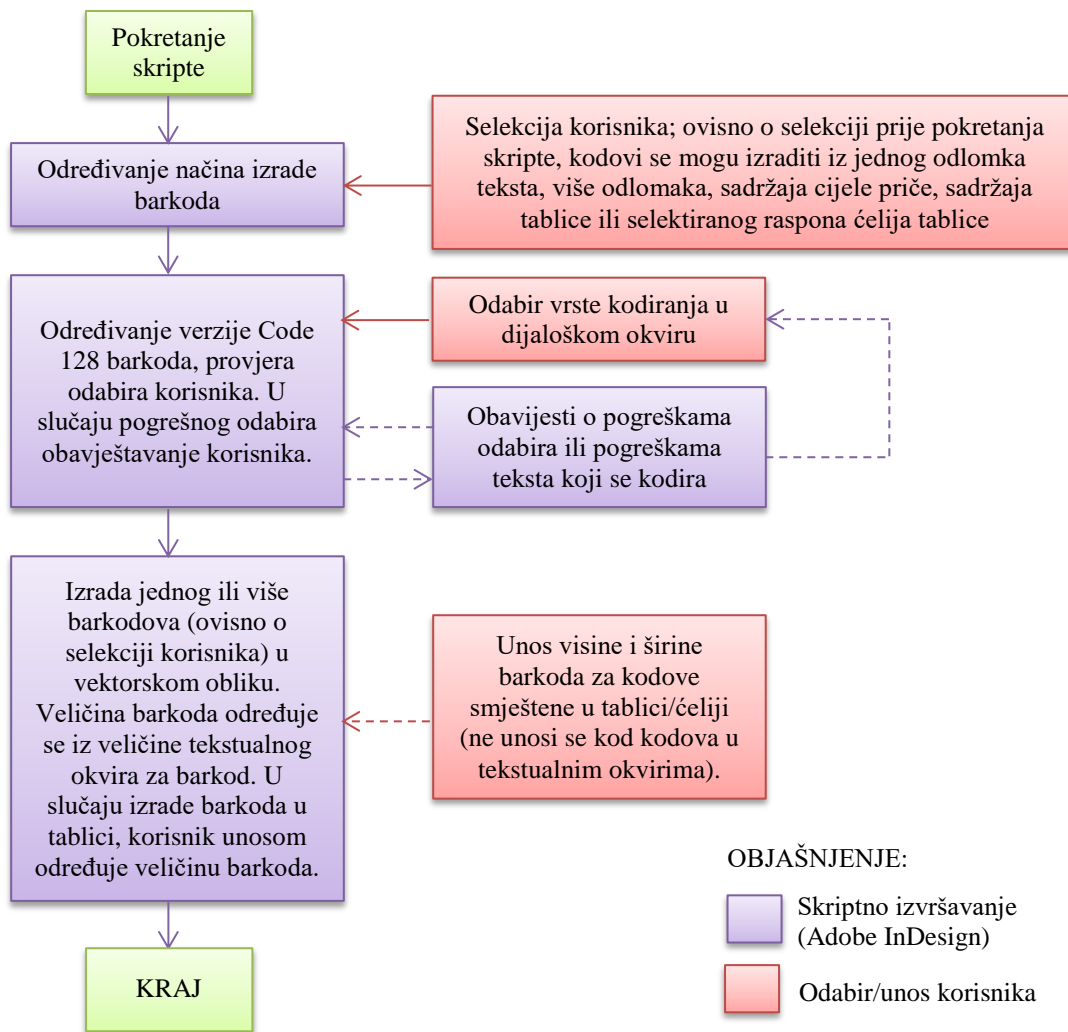
Slika 41. Izgled montaže i numeracije nakon provođenja početne skripte



Slika 42. Mogućnosti odabira verzija Code 128 barkoda



Slika 43. Tijek izvođenja prve skripte koja uključuje izradu numeracije (opcionalno) i montaže kartica te dodavanje polja za barkod

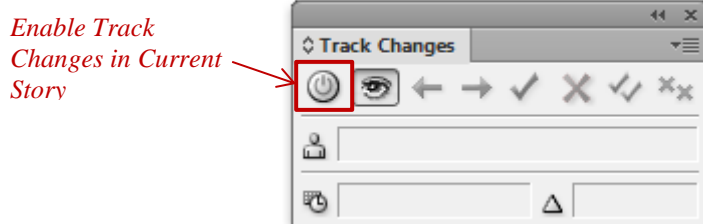


Slika 44. Tijek izvođenja druge skripte koja uključuje izradu Code 128 barkoda iz sadržaja tekstualnih okvira, tablica ili selektiranog dijela teksta

4.3.10. Skripta za obilježavanje ispravaka u tekstu nakon lekture (oznaka: S10)

Napisana skripta dodaje novu mogućnost u Adobe InDesign koja nije postojala ranije, a to je obilježavanje ispravaka lekture u prelomljenom tekstu. Adobe InDesign omogućuje praćenje ispravaka u tekstu ukoliko je uključena opcija *Track Changes* za pojedinu priču. Uključivanje navedene opcije omogućeno je preko gumba *Enable Track Changes in Current Story* palete *Track Changes* (prikazano na Slici 45.). Ispravke je moguće vidjeti jedino u *Story Editor* pregledu dokumenta. Izgled zabilježenih ispravaka prikazan je na Slici 46. Ispravke u *Story Editor* pregledu mogu vidjeti samo korisnici koji posjeduju program. Ispravke koji su izrađeni nije moguće prikazati u pregledniku stranica programa zajedno s formatiranjem teksta te ih nije moguće izvoziti iz programa kako bi se elektroničkim putem mogli proslijediti naručiocu na odobrenje.

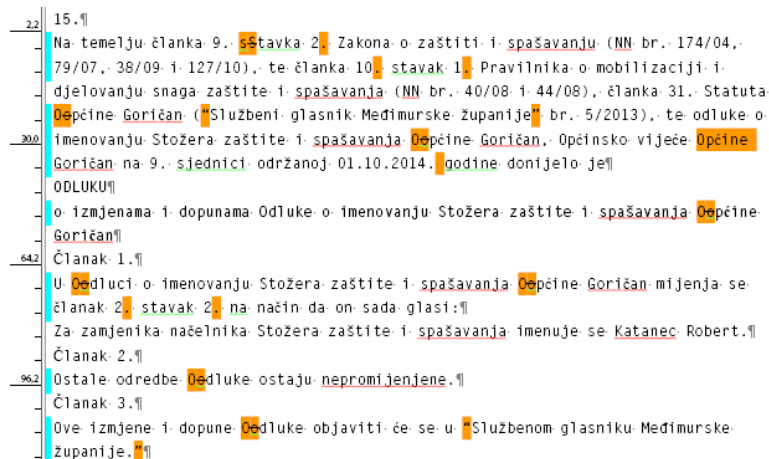
Iz navedenih razloga tipičan tijek zadataka nakon lekture je unošenje ispravaka u program. Naručiocu se tada na odobrenje šalje ispisana verzija dokumenta s ručno naznačenim ispravicima lekture te ispravljena verzija dokumenta za objavu. Odobrenje se teško vrši elektronski jer ispravke nije moguće prikazati u digitalnom obliku, osim ukoliko bi se skenirale sve stranice nakon lekture. Skeniranje stranica često nije prihvatljiva opcija zbog veličine skenirane datoteke i teškoće slanja elektronskim putem. Za skeniranje dokumenta s mnogo stranica također je potrebno utrošiti dodatno vrijeme i resurse poduzeća pa je jednostavnija opcija dostava ispisanih dokumenata.



Slika 45. *Track Changes* paleta programa Adobe InDesign

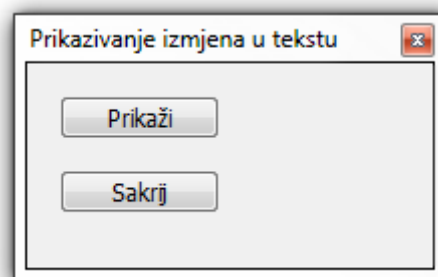
Izrađena skripta izmijenila je proces odobrenja ispravaka nakon lekture. Skripta je izrađena kako bi se izrađeni ispravci vidljivi u *Story Editor* paleti prikazivali i na stranicama dokumenta. Vidljive ispravke moguće je izvoziti u PDF dokument za elektronsko slanje na odobrenje. Skripta omogućuje prikazivanje i skrivanje ispravaka u pregledu prijeloma pa je uz ispravke moguće vidjeti i oblikovanje stranica za tisak. Izrađena skripta ima vrlo jednostavan izbornik s dva gumba. Slika 47. prikazuje dijaloški okvir skripte. Slika 48. prikazuje izgled teksta u dva PDF dokumenta koji se šalju naručitelju. Lijevi dio slike je dio dokumenta u

kojem se nalaze naznačeni ispravci, a desni dio slike je pregled dijela dokumenta za objavu. Skripta bojama preko opcije *Conditional Text* obilježava dio teksta koji je izbačen i dio teksta koji je umetnut. Izbačeni dijelovi prikazani su u narančastoj boji i obilježeni opcijom precrtavanja teksta, a umetnuti dijelovi u zelenoj boji što je vidljivo na Slici 49.



Slika 46. Pregled izrađenih ispravaka u tekstu nakon lekture u *Story Editor* pregledniku

Praćenje ispravaka u tekstu omogućuje se uključanjem *Track Changes* opcije programa. Nakon izrađenih ispravaka pokreće se skripta koja ih prikazuje ili skriva. Izvoženje vidljivih ispravaka u PDF dokument moguće je postavljenjem opcije *Indicators* palete *Conditional Text* na *Show and Print*. Navedena postavka prikazana je na Slici 50. Tijek izvođenja skripte prikazuje Slika 51.



Slika 47. Izbornik skripte za prikazivanje i skrivanje ispravaka u tekstu

15. Na temelju članka 9. Stavka 2. Zakona o zaštiti i spašavanju (NN br. 174/04, 79/07, 38/09 i 127/10), te članka 10. stavak 1. Pravilnika o mobilizaciji i djelovanju snaga zaštite i spašavanja (NN br. 40/08 i 44/08), članka 31. Statuta Općine Goričan (Službeni glasnik Međimurske županije br. 5/2013), te odluke o imenovanju Stožera zaštite i spašavanja Općine Goričan, Općinsko vijeće Općine Goričan na 9. sjednici održanoj 01.10.2014. godine donijelo je

ODLUKU

o izmjenama i dopunama Odluke o imenovanju Stožera zaštite i spašavanja Općine Goričan

Članak 1.

U Odluci o imenovanju Stožera zaštite i spašavanja Općine Goričan mijenja se članak 2. stavak 2. na način da on sada glasi:

Za zamjenika načelnika Stožera zaštite i spašavanja imenuje se Katanec Robert.

Članak 2.

Ostale odredbe Odluke ostaju nepromijenjene.

Članak 3.

Ove izmjene i dopune Odluke objavit će se u Službenom glasniku Međimurske županije

15. Na temelju članka 9. stavka 2. Zakona o zaštiti i spašavanju (NN br. 174/04, 79/07, 38/09 i 127/10), te članka 10. stavak 1. Pravilnika o mobilizaciji i djelovanju snaga zaštite i spašavanja (NN br. 40/08 i 44/08), članka 31. Statuta Općine Goričan (Službeni glasnik Međimurske županije br. 5/2013), te odluke o imenovanju Stožera zaštite i spašavanja Općine Goričan, Općinsko vijeće Općine Goričan na 9. sjednici održanoj 01.10.2014. godine donijelo je

ODLUKU

o izmjenama i dopunama Odluke o imenovanju Stožera zaštite i spašavanja Općine Goričan

Članak 1.

U Odluci o imenovanju Stožera zaštite i spašavanja Općine Goričan mijenja se članak 2. stavak 2. na način da on sada glasi:

Za zamjenika načelnika Stožera zaštite i spašavanja imenuje se Katanec Robert.

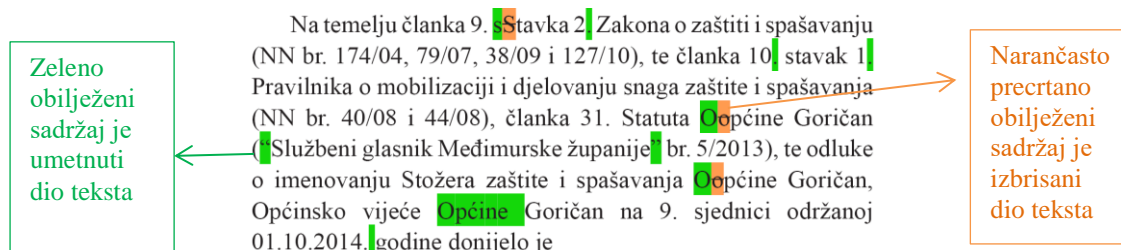
Članak 2.

Ostale odredbe Odluke ostaju nepromijenjene.

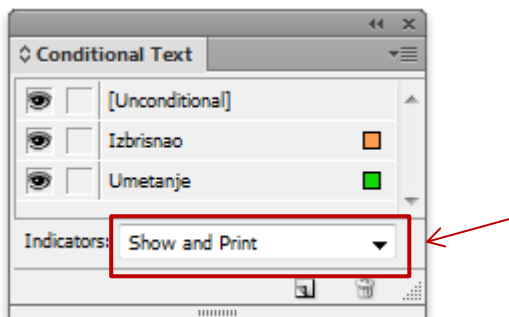
Članak 3.

Ove izmjene i dopune Odluke objavit će se u Službenom glasniku Međimurske županije."

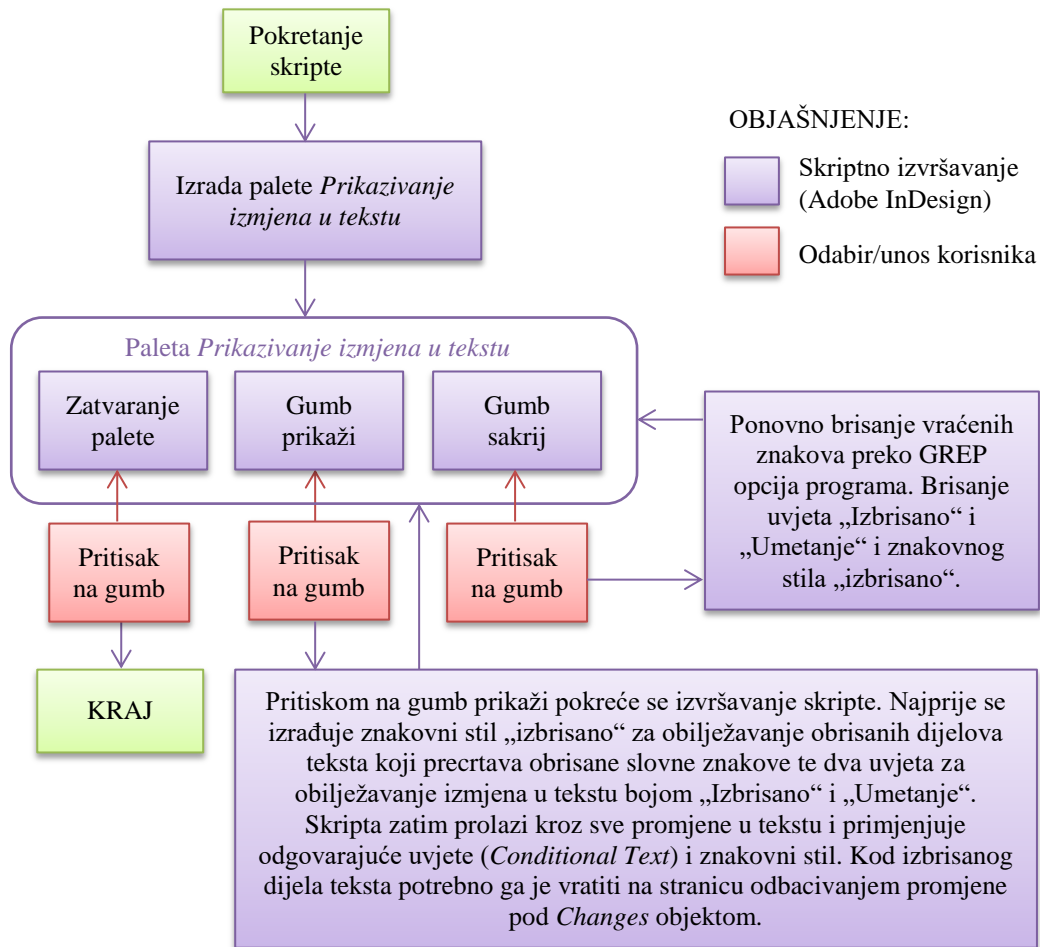
Slika 48. Izgled dijelova dokumenata koji se elektronski šalju na odobrenje naručitelju nakon lekture. Lijevi dio slike prikazuje ispravke, a desni dio slike prikazuje dokument sa skrivenim ispravicima



Slika 49. Prikaz ispravaka izrađenih u tekstu



Slika 50. Postavljanje *Show and Print* opcije *Conditional Text* palete za izvoženje ispravaka u PDF dokument



Slika 51. Tijek izvođenja skripte za prikazivanje ispravaka

4.4. Mjerenje optimizacije i obrada rezultata

Treća faza izrade rada uključuje vrednovanje izvedenih modela. Slijedom istraživanja, izvedeni prototipni modeli eksperimentalno su provjeravani i vrednovani u praksi. Prototipni modeli dani su odabranim ispitanicima sličnog predznanja na području grafičke pripreme koji su odradili zadatke na standardan i skriptni način. Iz eksperimentalnog rada potrebno je odrediti optimizaciju u vremenu izrade, automatizaciji i pouzdanosti.

4.4.1. Mjerenje vremena izvršavanja zadataka grafičke pripreme standardnim i skriptnim načinom

Ubrzanje grafičke pripreme mjereno je bilježenjem vremena. Rađena je usporedba standardnog i skriptnog vremena kako bi se odredila ušteda. Standardno vrijeme zadatka određeno je prosječnim vremenom 5 mjerenja s odabranim ispitanicima, a skriptno vrijeme određivanjem prosjeka vremena trajanja 5 izvođenja skripte. Unaprjeđenje vremena izvođenja zadataka grafičke pripreme iskazano je kao ukupna vremenska ušteda standardnog i skriptnog procesa u sekundama te kao ušteda vremena u postotku po pojedinom modelu. Mjerenja su rađena bilježenjem startnog i završnog vremena izvođenja zadataka.

Kako je jedan od zadataka proučavanja skriptnih modela moguća optimizacija skripti te određivanje njihovog optimalnog korištenja, radi točnijih je mjerenja u određenim slučajevima bilo korišteno skriptno mjerenje vremena izvršavanja skripte. Pritom je korištena skriptna opcija „*\$.hireTimer*“ koja omogućuje mjerenje vremena izvršavanja u mikrosekundama [57].

4.4.2. Mjerenje automatizacije izvršavanja zadataka grafičke pripreme skriptnim izvođenjem

Automatizacija procesa prikazana je brojem radnji koje je potrebno izvršiti standardnim načinom izvođenja zadatka prema broju koraka koji su potrebni pri izvođenju zadatka korištenjem skripte. Koraci su brojani kao pritisci tipki miša i tipkovnice programom „WhatPulse 2.3.1“¹⁹. Kod određivanja automatizacije uziman je prosječan broj pritisaka tipki miša i tipkovnice iz 5 mjerenja standardnog i skriptnog postupka i u postotku je iskazana automatizacija procesa. Navedenim mjerenjima želi se kvantificirati automatizacija procesa

¹⁹<http://www.whatpulse.org/>

izrade prijeloma korištenjem skripti. Provođenje skripte trebalo bi osigurati manji broj potrebnih radnji korisnika DTP programa za razliku od standardnih postupaka. Uvelike manji broj koraka očekuje se kod skripti koje mijenjaju kompleksne zadatke jer će oni uz pomoć skripte biti izrađeni automatizirano. Skripte odrađuju ponavljajuće zadatke, dok grafički dizajner ostaje zadužen za kreativna rješenja u prijelomu publikacija.

4.4.3. Mjerenje pouzdanosti izvršavanja zadataka grafičke pripreme skriptnim izvođenjem u usporedbi sa standardnim izvođenjem

Pouzdanost novih modela određuje se brojem pogrešaka u digitalnoj pripremi prilikom procesa pripreme dokumenata za tisak na standardan način i skriptnom izradom. Pogreškom se smatra pogreška u prijelomu koja ostaje neispravljena prije korekture. Dakle, pouzdanost je moguće iskazati kao broj pogrešaka za skriptne i standardne procese. Osim pogrešaka u prijelomu, greškom se može smatrati i povećani broj koraka prema najmanjem broju koraka za izradu operacije. Iz testiranja standardnog i skriptnog izvršenja zadatka moguće je izračunati pouzdanost obiju metoda rada.

Pogreške u interakciji čovjeka s računalom neizbježne su te istraživanja pokazuju da stručnjaci u radu u određenim programima imaju iznenađujuće veliku stopu pogreške, čak do 20 %. Pritom je kvalitativno moguće izdvojiti dvije vrste pogrešaka. Prva su vrsta pogreške koje nastaju kod korisnika koji izvode nove zadatke i ne uspijevaju otpočetak otkriti pravilan redoslijed akcija. Drugi tip pogrešaka su slučajne pogreške gdje stručnjaci imaju ispravnu namjeru, ali ne uspijevaju ispravno izvesti određeni redoslijed akcija. Dosadašnji eksperimenti pokazuju da su stope pogreški za stručnjake između 5 i 20 %. U svim istraživanjima stručnjaka koji obavljaju određene zadatke utvrdilo se da stručnjaci na kraju ipak izvedu ispravne rezultate. Većina spomenutih pogrešaka slučajne su pogreške u akcijama. Oko 50 % učinjenih pogrešaka detektira se nakon njihovog nastajanja i ispravlja. Otkrivanje i ispravljanje pogrešaka dio je vještina stručnjaka [58].

Iz navedenih su razloga u istraživanju i usporedbi pouzdanosti standardnog i skriptnog izvršavanja zadatka svi ispitanici bili najprije upućeni u istu proceduru rješavanja zadatka. Važno je napomenuti da se pojedini zadatci u programima za prijelom mogu različito obavljati, različitim redoslijedom i upotrebom različitih alata pa bi bilo teško uspoređivati različite procedure. Pri svakom standardnom mjerenju pokušalo se primijeniti najbolje rješavanje zadatka, u smislu najmanjeg broja potrebnih koraka i najkraćeg vremena obavljanja zadatka.

Pouzdanost skripte gledana s aspekta broja pogrešaka koje ostaju neispravljene iznimno je velika. Ukoliko je skripta dovoljno testirana i optimizirana, vjerojatnost je pogreške izuzetno mala. Za sve skripte testirane u ovom radu ni u jednom testu nije došlo do pogreške u izvršenju skripte. Utjecaj skripte na pouzdanost izvršavanja zadatka moguće je zanemariti jer je ona načelno vezana uz pouzdanost čovjeka i njegovu stopu pogrešaka u skriptnom izvršavanju zadatka. Pogreške u izvršavanju skripte mogle bi se dogoditi ukoliko korisnik odredi pogrešne ulazne parametre, dakle veća je vjerojatnost da će pogreška nastati zbog ljudskog faktora.

Pouzdanosti i varijacije u broju potrebnih koraka pri izvođenju zadatka za svaku skriptnu i standardnu izradu kod testiranih skripti prikazane su sljedećim vrijednostima i računane preko niže navedenih formula:

- **Raspon varijacije**

$$RV = b_{\max} - b_{\min} \quad (1)$$

- **Prosjek**

$$\bar{b} = \sum_{i=1}^5 (b_i) / 5 * 100\% \quad (2)$$

- **Standardna devijacija**

$$\sigma = \sqrt{\frac{\sum_{i=1}^5 (b_i - \bar{b})^2}{n}} \quad (3)$$

- **Koeficijent varijacije**

$$V = \frac{\sigma}{\bar{b}} * 100\% \quad (4)$$

- **Najveće odstupanje od minimalne vrijednosti**

$$V_{\max} = \frac{b_{\max} - b_{\min}}{b_{\min}} * 100\% \quad (5)$$

- **Prosječno odstupanje od minimalne vrijednosti**

$$\bar{V} = \sum_{i=1}^5 (b_i - b_{\min}) / 5 * 100\% \quad (6)$$

gdje su:

RV = raspon varijacije broja pritisaka tipki miša i tipkovnice skriptnog ili standardnog izvođenja zadatka

b_{\min} = minimalan broj pritisaka tipki miša i tipkovnice skriptnog ili standardnog izvođenja zadatka

b_{\max} = maksimalan broj pritisaka tipki miša i tipkovnice skriptnog ili standardnog izvođenja zadatka

σ = standardna devijacija broja potrebnih koraka skriptnog ili standardnog izvođenja zadatka

\bar{b} = prosječna vrijednost broja pritisaka tipki miša i tipkovnice skriptnog ili standardnog izvođenja zadatka

b_i = broj pritisaka tipki miša i tipkovnice pojedinačnog i-tog mjerenja skriptnog ili standardnog izvođenja zadatka

V = koeficijent varijacije broja pritisaka tipki miša i tipkovnice skriptnog ili standardnog izvođenja zadatka

V_{\max} = najveće odstupanje od minimalne vrijednosti broja pritisaka tipki miša i tipkovnice skriptnog ili standardnog izvođenja zadatka

\bar{V} = prosječno odstupanje broja pritisaka tipki miša i tipkovnice od minimalne vrijednosti skriptnog ili standardnog izvođenja zadatka.

Za određivanje optimalnog rješavanja zadatka uzima se najmanji izmjereni broj koraka, tj. pritisaka tipki miša i tipkovnice. Za prikaz pouzdanosti pojedine skripte uzeto je prosječno odstupanje od minimalne vrijednosti kao stopa pogrešaka. Ukoliko se uzme minimalan broj pritisaka miša i tipkovnice, može se izračunati:

$$P(\text{standardno ili skriptno}) = (1 - Sp)^{Km} \quad (7)$$

gdje je:

Sp = prosječna stopa pogrešaka

Km = minimalan broj koraka

P (standardno ili skriptno) = vjerojatnost da će se zadatak ostvariti s minimalnim brojem koraka kod standardnog ili skriptnog izvođenja zadatka.

5. REZULTATI

5.1. Rezultati mjerenja skripte za brisanje neželjenih bjelina u tekstu (S1)

Tablice 4. i 5. prikazuju rezultate dobivene mjerenjem dok su usporedbe standardnog i skriptnog izvođenja zadataka prikazane na Slikama 52. i 53.

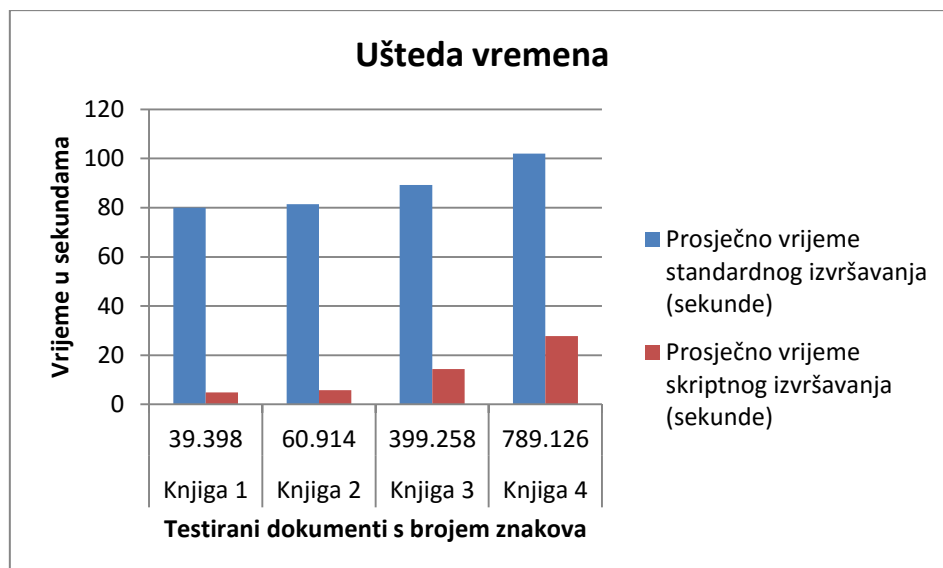
U testiranju je sudjelovalo 5 stručnjaka grafičke pripreme koji su zadatke izvršavali standardnim i skriptnim izvođenjem. Mjerenja su rađena na 4 različita dokumenta dobivena za prijelom od strane naručioca. Dokumenti su se razlikovali brojem znakova koje sadrže, kako se može vidjeti u Tablicama 4. i 5. Time se željela usporediti ušteda vremena na dokumentima različitih obima. Zadatci koji su se izvršavali skriptnim i standardnim izvođenjem opisani su u poglavlju 4.3.1. Pojedinačni rezultati mjerenja prikazani su pod Prilozima u Tablici 40.

Tablica 4. Usporedbe vremena skriptnog i standardnog izvođenja zadatka brisanja neželjenih bjelina

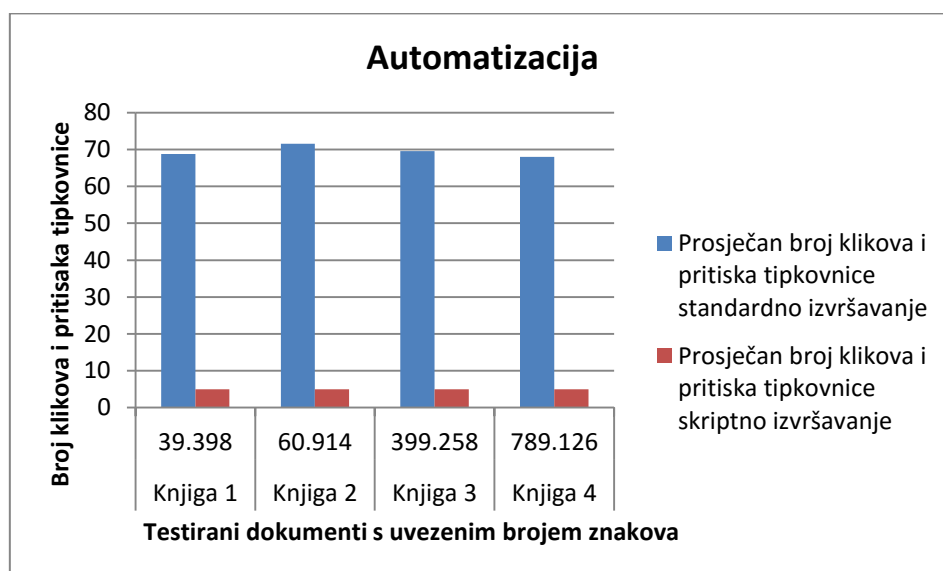
	Broj znakova uvezeni dokument	Broj znakova nakon zadatka	Broj znakova razlike	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postotci)
Knjiga 1	39.398	39.348	50	80	5	75	94 %
Knjiga 2	60.914	60.572	342	81	6	76	93 %
Knjiga 3	399.258	398.360	898	89	14	75	84 %
Knjiga 4	789.126	787.466	1.660	102	28	74	73 %

Tablica 5. Usporedbe broja klikova mišem i pritisaka tipkovnice skriptnog i standardnog izvođenja zadatka brisanja neželjenih bjelina

	Broj znakova uvezeni dokument	Broj znakova nakon zadatka	Broj znakova razlike	Prosječan broj klikova i pritiska tipkovnice standardno izvršavanje	Prosječan broj klikova i pritiska tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
Knjiga 1	39.398	39.348	50	69	5	93 %
Knjiga 2	60.914	60.572	342	72	5	93 %
Knjiga 3	399.258	398.360	898	70	5	93 %
Knjiga 4	789.126	787.466	1.660	68	5	93 %



Slika 52. Prikaz uštede vremena skriptnim izvođenjem zadatka brisanja neželjenih bjelina



Slika 53. Prikaz dobivene automatizacije skriptnim u usporedbi sa standardnim izvođenjem zadatka brisanja neželjenih bjelina

Iz rezultata se može zaključiti da je pri ovom zadatku dobivena automatizacija neovisna o veličini dokumenta, tj. broju znakova koje on sadržava. Vrijeme izvođenja je ovisno o veličini dokumenta i raste kako se povećava broj znakova u dokumentu. Značajniji rast vremena izvođenja vidljiv je kod skriptnog izvođenja zadatka dok kod standardnog načina veličina dokumenta ima manji utjecaj na vrijeme izvođenja. Razlog manjeg povećanja vremena kod standardnog načina izvođenja je jednak broj zadataka koje korisnik mora odraditi u svim testovima, neovisno o veličini dokumenta. Pri standardnom izvođenju zadataka najveći utjecaj na dobiveno vrijeme imaju radnje koje poduzima korisnik dok je

mного manji udio vremena koji je potreban programu da odradi *GREP* zadatke. Kako kod skriptnog izvršavanja korisnik nema utjecaj na brzinu izvođenja zadatka, vrijeme će manje varirati u pojedinačnim testovima, a veći će utjecaj na vrijeme izvođenja imati broj znakova koje je potrebno ispraviti.

U svim mjerenjima dobiven je jednak konačan rezultat. Ni u jednom izvršavanju zadatka nije došlo do varijacija u konačnom rezultatu, dakle svi su dokumenti bili ispravno pripremljeni.

Do varijacija u izradi dolazilo je prilikom standardnog izvođenja zadatka u vremenu izrade i broju potrebnih koraka. Iz Tablice 41. izvedeni su sumarni rezultati koji prikazuju varijacije u izvođenju zadataka i prikazani u Tablici 6.

Tablica 6. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka brisanja neželjenih bjelina

	Knjiga 1		Knjiga 2		Knjiga 3		Knjiga 4		Ukupno/ prosjek za sva mjerenja	
Broj znakova uvezeni dokument	39.398		60.914		399.258		789.126			
Broj znakova nakon zadatka	39.348		60.572		398.360		787.466			
Broj znakova razlike	50		342		898		1.660			
Izvršenje zadatka	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno
Pritisaci miša i tipkovnice										
Minimalan broj (b_{min})	66	5	65	5	65	5	67	5	65	5
Maksimalan broj (b_{max})	75	5	78	5	77	5	70	5	78	5
Raspon varijacije (RV)	9	0	13	0	12	0	3	0	9	0
Prosjek (\bar{b})	69	5	72	5	70	5	68	5	70	5
Standardna devijacija (σ)	4	0	5	0	4	0	1	0	3	0
Koeficijent varijacije (V)	5 %	0 %	6 %	0 %	6 %	0 %	2 %	0 %	5 %	0 %
Najveće odstupanje od minimalne vrijednosti (V_{max})	14 %	0 %	20 %	0 %	18 %	0 %	4 %	0 %	20 %	0 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	4 %	0 %	10 %	0 %	7 %	0 %	1 %	0 %	5 %	0 %

Za određivanje optimalnog rješavanja zadatka uzima se najmanji izmjereni broj koraka, tj. pritisaka miša i tipkovnice. Vidljivo je da broj koraka u svim standardno izvedenim zadacima značajno varira, čak do 20 %. Prosječno odstupanje od minimalnog broja koraka u svim zadacima iznosilo je 5 %. Iz navedenog je moguće zaključiti da je pouzdanost, tj.

vjerojatnost izvođenja zadatka na optimalan način, tj. s minimalnim brojem koraka vrlo niska kod standardnog načina izvođenja zadatka.

Kada uzmemo da je prosječno odstupanje od minimuma stopa pogrešaka koja iznosi 5 % i ukoliko uzmemo minimalan broj pritisaka miša i tipkovnice od 65, možemo izračunati prema formuli (7):

$$P(\text{standardno}) = (1 - 0,05)^{65} = 3,56 \%$$

Iz izračuna vjerojatnosti možemo zaključiti da je pouzdanost izvođenja zadatka s minimalnim brojem koraka niska i da će kod standardnog izvođenja zadatka vrlo vjerojatno doći do odstupanja u broju potrebnih koraka u izvođenju te će se zadatak rijetko izraditi s minimalnim brojem potrebnih koraka.

Kod skriptnog načina izvođenja zadatka nije došlo do odstupanja u izvođenju. Uvijek je bio potreban jednak broj koraka pa možemo izračunati vjerojatnost izvođenja zadatka s minimalnim brojem koraka:

$$P(\text{skriptno}) = (1 - 0)^5 = 100 \%$$

5.2. Rezultati mjerenja skripte za zadržavanje uvezenih znakovnih stilova (S2)

Testiranje obavljanja svakodnevnih zadataka zadržavanja uvezenih znakovnih stilova u dokumentu koji se prelama rađena su na standardan i skriptni način na 3 različita dokumenta uvezena u InDesign iz Word dokumenata dobivenih za prijelom. Dokumenti se razlikuju u broju sadržanih znakova. Obavljanje zadataka na standardan i skriptni način opisano je u Poglavlju 4.3.2., a dobiveni rezultati pojedinačnih mjerenja prikazani su u poglavlju Prilozi u Tablici 42. Tablice 7. i 8. te slike 54. i 55. prikazuju sumarne rezultate mjerenja.

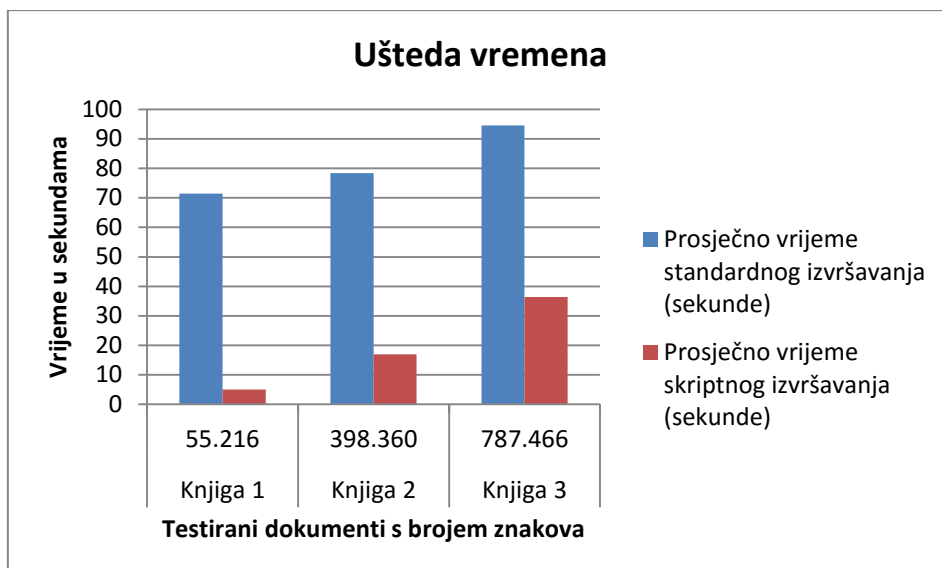
Tablica 7. Usporedbe vremena skriptnog i standardnog izvođenja zadatka zadržavanja uvezenih znakovnih stilova

	Broj znakova uvezeni dokument	Broj postavljanja stilova	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postoci)
Knjiga 1	55.216	146	71	5	66	93 %
Knjiga 2	398.360	1.166	78	17	61	78 %
Knjiga 3	787.466	2.246	95	36	58	62 %

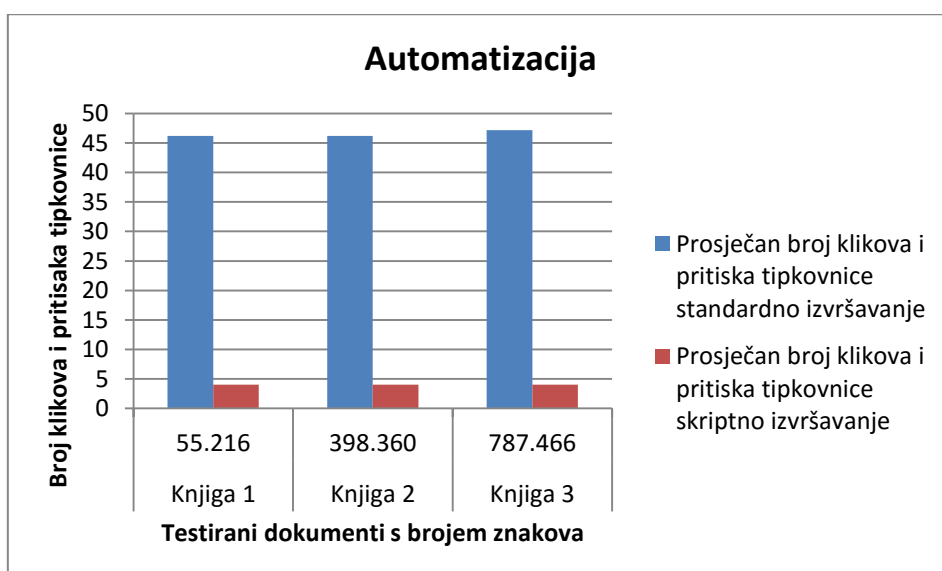
Tablica 8. Usporedbe broja klikova i pritiska tipkovnice skriptnog i standardnog izvođenja zadatka zadržavanja uvezenih znakovnih stilova

	Broj znakova uvezeni dokument	Broj postavljanja stilova	Prosječan broj klikova i pritiska tipkovnice standardno izvršavanje	Prosječan broj klikova i pritiska tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
Knjiga 1	55.216	146	46	4	91 %
Knjiga 2	398.360	1.166	46	4	91 %
Knjiga 3	787.466	2.246	47	4	92 %

Iz dobivenih rezultata može se zaključiti da se povećanjem broja znakova u dokumentu povećava i vrijeme potrebno za izvršenje zadatka. Povećanje broja znakova ima veći utjecaj na potrebno vrijeme kod skriptnog izvršavanja zadatka. Dobivena ušteda vremena u postocima ovisit će o broju znakova u dokumentu i ispravcima koji se obavljaju. Također se može usporediti dobivena ušteda u vremenu s testiranjem skripte S1 (poglavlje 5.1.) i zaključiti da će se značajnije uštede u vremenu skriptnim izvođenjem zadatka dobiti kod poslova koji su dugotrajniji i zahtijevaju više slijednih koraka.



Slika 54. Prikaz uštede vremena skriptnim izvođenjem zadatka zadržavanja uvezenih znakovnih stilova



Slika 55. Prikaz dobivene automatizacije skriptnim izvođenjem zadatka zadržavanja uvezenih znakovnih stilova

Pri izvođenju zadatka na standardan i skriptni način nije došlo od pogrešaka koje su ostale neispravljene. Pogreške su se ispravljale u trenutku njihovog nastanka pri standardnom izvođenju zadatka tako da i u testiranju ove skripte postoje varijacije u interakciji korisnika sa sučeljem programa. Tablica 9. prikazuje varijacije pri izvođenju zadatka. Iz dobivenih je rezultata vidljivo da testirano standardno izvođenje zadatka ima niže varijacije u broju potrebnih koraka za izvođenje zadatka u usporedbi s rezultatima pod poglavljem 5.1. Manja varijacija u izvođenju zadatka dobivena je zbog manjeg broja koraka potrebnih za standardno

izvršavanje zadatka. Ukoliko izračunamo pouzdanost prema formuli (7) s ukupnim podacima iz Tablice 9., dobivamo:

$$P(\text{standardno}) = (1 - 0,03)^{45} = 25,39 \%$$

$$P(\text{skriptno}) = (1 - 0)^4 = 100 \%$$

Tablica 9. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka zadržavanja znakovnih stilova

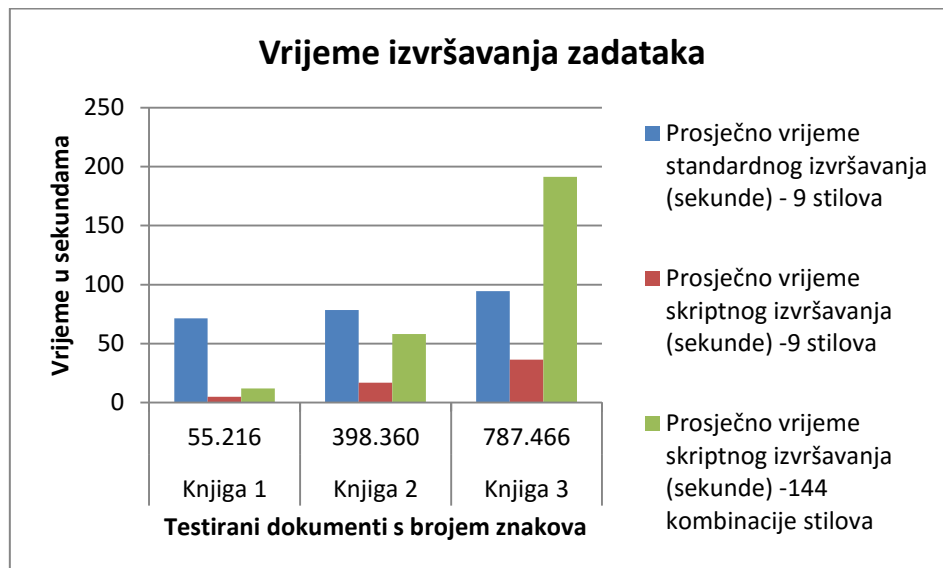
	Knjiga 1		Knjiga 2		Knjiga 3		Ukupno/ prosjek za sva mjerenja	
Broj znakova uvezeni dokument	55.216		398.360		787.466			
Broj postavljanja stilova	146		1.166		2.246			
Izvršenje zadatka	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno
Pritisci miša i tipkovnice								
Minimalan broj (b_{\min})	45	4	45	4	46	4	45	4
Maksimalan broj (b_{\max})	48	4	47	4	48	4	48	4
Raspon varijacije (RV)	3	0	2	0	2	0	3	0
Prosijek (\bar{b})	46	4	46	4	47	4	47	4
Standardna devijacija (σ)	1	0	1	0	1	0	1	0
Koeficijent varijacije (V)	3 %	0 %	2 %	0 %	2 %	0 %	2 %	0 %
Najveće odstupanje od minimalne vrijednosti (V_{\max})	7 %	0 %	4 %	0 %	4 %	0 %	7 %	0 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	3 %	0 %	3 %	0 %	3 %	0 %	3 %	0 %

Pouzdanost možemo zaključiti da je vjerojatnost izvršavanja zadatka s minimalnim brojem koraka kod standardnog izvršenja zadatka veća ukoliko imamo manji broj koraka koje je potrebno izraditi.

Testirana skripta i standardan način izvođenja zadatka provjeravaju pojavljivanje samo 9 znakovnih stilova koji bi se mogli pojaviti u dokumentu. Time je zanemarena mogućnost pojavljivanja kombinacija stilova. Kako bi se uključila i mogućnost kombinacije znakovnih stilova, izrađena je skripta koja pronalazi 9 osnovnih znakovnih stilova i sve njihove kombinacije što u konačnici daje 144 testa. Izrađena skripta prema potrebi kreira nove znakovne stilove. Izvođenje zadatka traženja 144 pojavljivanja kombinacije stilova standardnim načinom izvođenja bilo bi neisplativo i teško ostvarivo. Skriptnim načinom izvođenja ovakva je mogućnost izvediva kao što je dokazano izrađenom skriptom. Napisana skripta koja provjerava sve kombinacije pojavljivanja stilova, je preciznija od testiranih

načina rada i daje prednosti u usporedbi sa standardnim izvođenjem u smislu da omogućuje radnje koje su teško ostvarive, a daju veću pouzdanost izrade kvalitetne pripreme.

Skripta za traženje svih kombinacija stilova omogućuje zadržavanje svih lokalnih promjena nakon primjene stilova odlomaka jer će svaka izmjena biti zabilježena kao ispravan znakovni stil. Upravo zbog dodanih zadataka koje skripta obavlja vrijeme izvršavanja će biti duže. Dobivena automatizacija jednaka je ranije testiranoj skripti. Rezultati mjerenja skripte prikazani su u poglavlju Prilozi u Tablici 43. Tamo su prikazane i dodatne kombinacije pronađene u testiranim dokumentima. Slika 56. prikazuje usporedbu prosječnih vremena izvršavanja zadatka u usporedbi s početnom skriptom i standardnim načinom izvršavanja zadatka.



Slika 56. Prikaz vremena skriptnog izvođenja zadatka za zadržavanje znakovnih stilova (9 i 144 kombinacije stilova) u usporedbi sa standardnom izradom

Iz Slike 56. vidljivo je da će skripta koja traži i kombinacije stilova donijeti uštede u vremenu izvođenja kod manjeg broja znakova u dokumentu prema standardnom izvođenju. Kod velikog broja znakova u dokumentu vrijeme izvršavanja će biti veće, ali zanemarivo prema prednostima koje provođenje navedene skripte donosi pripremi dokumenta. Testiranje standardne izrade pripreme koja uključuje traženje 144 kombinacije stilova nije rađeno zbog neisplativosti. Vrijeme koje bi se utrošilo za konačnu ispravnost pripreme nije isplativo i ovakva priprema se gotovo nikada ne bi odrađivala standardnim načinom.

5.3. Rezultati mjerenja skripte za uređivanje tablica u periodičkoj publikaciji (S3)

Testiranje obavljanja zadataka uređenja tablica proračuna u periodičkoj publikaciji rađena su na standardan i skriptni način na 3 različite tablice proračuna uvezene u InDesign iz Excel dokumenta dobivenog za prijelom. Dokumenti se razlikuju u broju redaka tablice. Obavljanje zadataka na standardan i skriptni način opisano je u poglavlju 4.3.3. Dobiveni rezultati pojedinačnih mjerenja prikazani su u poglavlju Prilozi u Tablicama 44. i 45.

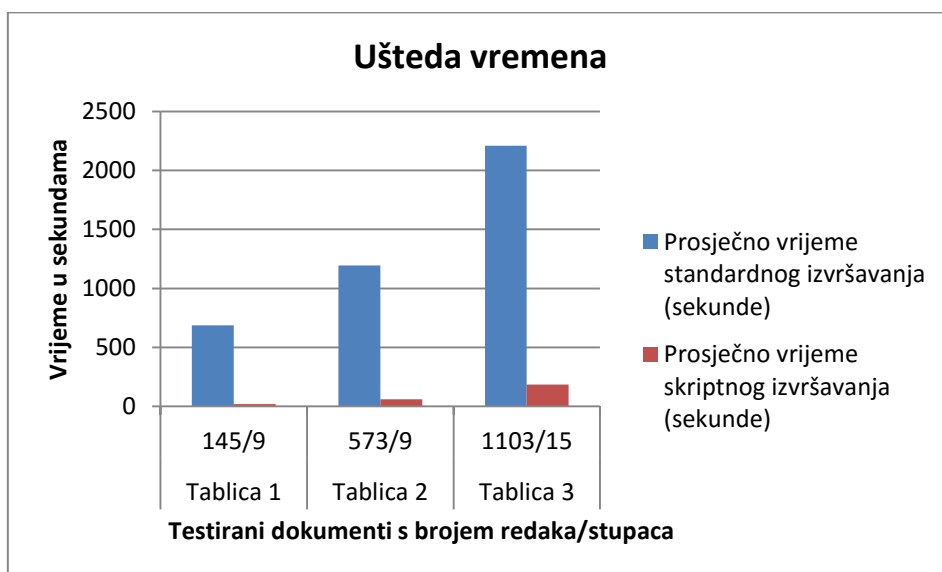
Iz Tablice 44. vidljivo je da je došlo do neispravljenih pogrešaka kod standardnog načina izvođenja zadatka. Kod skriptnog načina izvođenja zadatka kasnija lektura nije pronašla nijednu pogrešku (Tablica 45.) što ukazuje na činjenicu da je skriptno rješavanje zadatka pouzdanije. U prosjeku su se dogodile 3 pogreške u prijelomu koje su ispravljene nakon lekture. Samo je jedan od 15 standardno rađenih prijeloma dao u potpunosti ispravno prelomljen dokument, tj. početno je 93 % dokumenata bilo neispravno prelomljeno što je iznimno velik postotak pogrešaka. Tako velik postotak pogrešaka dobiven je zbog kompleksnosti zadatka. To možemo zaključiti usporedbom s ranije rađenim zadacima u poglavljima 5.1. i 5.2. gdje su se odrađivali jednostavniji zadaci s mnogo manjim brojem koraka te ni u jednom testu nije došlo do pogreške.

Tablica 10. Usporedbe vremena skriptnog i standardnog izvođenja zadatka uređivanja tablica u periodičkoj publikaciji

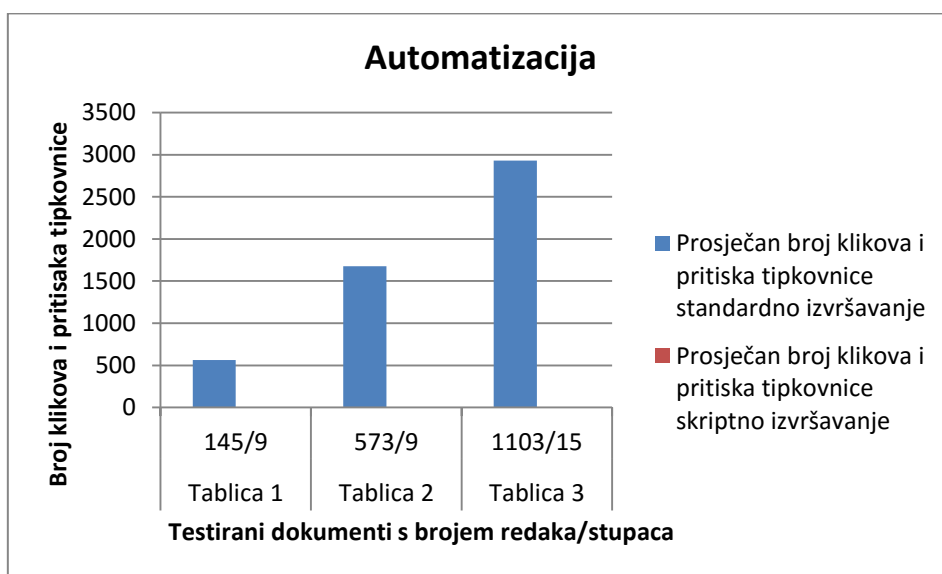
	Broj redaka/stupaca uvezene tablice	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postoci)
Tablica 1	145/9	687	21	666	97 %
Tablica 2	573/9	1194	59	1135	95 %
Tablica 3	1103/15	2210	186	2024	92 %

Tablica 11. Usporedbe broja klikova i pritiska tipkovnice skriptnog i standardnog izvođenja zadatka uređivanja tablica u periodičkoj publikaciji

	Broj redaka/stupaca uvezene tablice	Prosječan broj klikova i pritiska tipkovnice standardno izvršavanje	Prosječan broj klikova i pritiska tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
Tablica 1	145/9	564	4	99 %
Tablica 2	573/9	1675	4	100 %
Tablica 3	1103/15	2932	4	100 %



Slika 57. Prikaz uštede vremena skriptnim izvođenjem zadatka uređivanja tablica u periodičkoj publikaciji



Slika 58. Prikaz dobivene automatizacije skriptnim izvođenjem zadatka uređivanja tablica u periodičkoj publikaciji

Sumarni rezultati mjerenja koji uključuju i popravke prikazani su u Tablicama 10. i 11., a usporedbe na Slikama 57. i 58. Iz testiranja se može zaključiti da se kod skripti koje izvršavaju kompleksne i dugotrajne zadatke dobiva iznimno visok stupanj automatizacije i velika ušteda u vremenu.

Kod izračuna vjerojatnosti da će se zadatak ostvariti s minimalnim brojem koraka kod standardnog rješavanja zadatka može se zaključiti da je vjerojatnost minimalnog opterećenja

operatera, tj. vjerojatnost da pri rješavanju zadatka neće doći do pogrešaka u izvođenju izuzetno mala (gotovo jednaka nuli), a stopa pogrešaka iznimno velika. Tablica 12. prikazuje varijacije u izvođenju zadatka.

Prema formuli (7) moguće je izračunati pouzdanosti:

$$P(\text{standardno}) = (1 - 0,19)^{1724} = 0 \%$$

$$P(\text{skriptno}) = (1 - 0)^4 = 100 \%$$

Tablica 12. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka uređivanja tablica u periodičkoj publikaciji

	Tablica 1		Tablica 2		Tablica 3		Ukupno/ prosjek	
Broj redaka/stupaca uvezene tablice	145/9		573/9		1103/15			
Izvršenje zadatka	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno
Pritisaci miša i tipkovnice								
Minimalan broj (b_{\min})	511	4	1455	4	2209	4	-	-
Maksimalan broj (b_{\max})	644	4	1794	4	3454	4	-	-
Raspon varijacije (RV)	133	0	339	0	1245	0	-	-
Prosjek (\bar{b})	564	4	1675	4	2932	4	1724	4
Standardna devijacija (σ)	56	0	131	0	506	0	-	-
Koeficijent varijacije (V)	10 %	0 %	8 %	0 %	17 %	0 %	12 %	0 %
Najveće odstupanje od minimalne vrijednosti (V_{\max})	26 %	0 %	23 %	0 %	56 %	0 %	35 %	0 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	10 %	0 %	15 %	0 %	33 %	0 %	19 %	0 %

Iz dobivenih rezultata skripti S1, S2 i S3 vidljivo je da se može dobiti visoka vjerojatnost ostvarenja zadatka s minimalnim brojem potrebnih koraka bez obzira radi li se o jednostavnom ili kompleksnom zadatku. Također se može zaključiti da dobivena automatizacija i minimalan broj potrebnih koraka kod skriptnog rješavanja zadatka ne ovise o kompleksnosti zadatka koji se izvodi. U sve tri ranije navedene skripte potreban je gotovo jednak broj klikova koji nimalo ne variraju u ovisnosti o izvršitelju zadatka.

Kod standardnog izvođenja zadatka imamo veća odstupanja od minimuma kako raste broj zadataka koji se obrađuju. Vjerojatnost pogreške u izvođenju sve više raste, kao i vjerojatnost da će nakon izvođenja zadatka ostati neispravno prelomljeni dijelovi dokumenta. Standardno rađena grafička priprema uvelike ovisi o stručnosti osobe koja izvršava traženi zadatak. Upotrebom skriptnih tehnologija navedeni problemi mogu se ukloniti.

5.4. Rezultati mjerenja skripte za izradu numeracije i montaže kartica (S4)

Skripta za izradu numeracije i montaže, osim za mjerenje uštede vremena, automatizacije i pouzdanosti, korištena je i za testiranje mogućnosti optimizacije skripti za kraće vrijeme izvršavanja.

Sva mjerenja optimizacije vremena izvođenja rađena su izradom montaže istog PDF dokumenta. Budući da je skripta dinamička te izvođenje ovisi o odabiru korisnika, mjerenja su rađena sa sljedećim fiksnim postavkama: veličina montažnog arka 450 x 320 mm; veličina PDF dokumenta 2,32 MB; veličina pojedine PDF kartice na montaži 85 x 54 mm; napusti 3 mm, *Opcija montaže* postavljena na *Glava gore: 20 na arku* koja omogućuje montažu od 20 kartica na arku; numeracija bez vodećih nula, *Opcije numeracije* postavljene na *Kao blok, numeracija se povećava kroz arke* (za izradu blokova sa slijednim brojevima) s rasponom numeracije od 1 do 1000.

Tijek izvršavanja skripte i njene mogućnosti opisane su u poglavlju 4.3.4.

5.4.1. Optimizacija skripte za kraće vrijeme izvođenja

Na vrijeme izvođenja skripte pri njenom izvršavanju utječe korisnik. Korisnik nema utjecaja samo na vrijeme izvršavanja numeracije i montaže pa su navedena vremena testirana pri određivanju optimizacije skripte za brže izvođenje. Vrijeme potrebno za izvršenje zadatka koje uključuje i vrijeme utrošeno na zadatke koje izvodi korisnik uključeno je jedino u posljednjem mjerenju koje uspoređuje standardno i skriptno vrijeme izvođenja zadatka.

Vremena prikazana u dijelu određivanja optimizacije same skripte prosjek su 5 mjerenja skripte. Skriptno vrijeme dano je kao vrijeme montaže, numeriranja i vrijeme čekanja.

Vrijeme čekanja (eng. *idle time*) je vrijeme koje prođe nakon izvršenja skripte do trenutka kada korisnik može nastaviti s radom u programu (uključuje vrijeme iscrtavanja ekrana i ostale zadatke koje izvodi program nakon izvršenja skripte). Vrijeme čekanja mjereno je skriptnim načinom pomoću slušatelja događaja (eng. *event listener*). InDesign *idle* zadatci izvršavaju se kada ne postoje drugi zadatci u poretku rješavanja zadataka samog programa [29].

5.4.1.1. Početna vremena izvođenja skripte

Tablica 13. prikazuje rezultate mjerenja originalno napisane skripte. Varijacije pri izvođenju skripte rađene su u broju tekstualnih okvira po kartici koji sadrže numeraciju. Tablica prikazuje prosječna vremena pet mjerenja za svaku opciju.

Tablica 13. Vrijeme izvršavanja originalne skripte u sekundama

Broj okvira za numeraciju po kartici	1	2	3	4	5
Vrijeme montaže	0,82	0,95	1,14	1,25	1,40
Vrijeme numeracije	30,84	73,46	127,44	196,22	273,55
Vrijeme čekanja	26,14	51,09	112,91	192,65	323,28
Ukupno vrijeme	57,80	125,50	241,49	390,12	598,23

Iz rezultata u Tablici 13. može se zaključiti da je vrijeme montaže kratko i da se ne mijenja značajno dodavanjem većeg broja okvira za numeraciju po kartici. Vrijeme numeracije i vrijeme čekanja najduža su vremena pri izvršavanju skripte te je pretpostavljeno da će se upravo navedena vremena moći optimizirati.

5.4.1.2. Testiranje *doScript* metode postavljanjem *UndoModes* parametra funkcije

Kako bi se odredio utjecaj *doScript* metode i njenog opcionalnog parametra *UndoModes*, pri mjerenju je odabrano izvršenje skripte s ranije navedenim postavkama. Posebno je postavljena opcija *Brojpolja za numeraciju po kartici* na 2 okvira.

Podešavanje *UndoModes* opcije *doScript* metode jedna je od rijetkih optimizacija koja se preporučuje u literaturi. *UndoModes* parametar određuje kako će program zapisivati *Undo*²⁰ opciju pri izvršenju skripte. Adobe tvrdi: „InDesign omogućuje bilježenje *Undo* koraka gotovo svake akcije, ali to dolazi pod cijenom: za svaku akciju koju učinite, InDesign zapisuje na disk. Alati koje je moguće odabirati u korisničkom sučelju pri normalnim operacijama ne predstavljaju nikakav problem. Za skripte koje mogu izvršavati tisuće akcija u vremenu u kojem čovjek može trepnuti, konstantno pristupanje disku može izazvati ozbiljnu prepreku na performanse. *doScript* metoda nudi način zaobilazanja navedenog uskog grla

²⁰ Poništavanje koraka

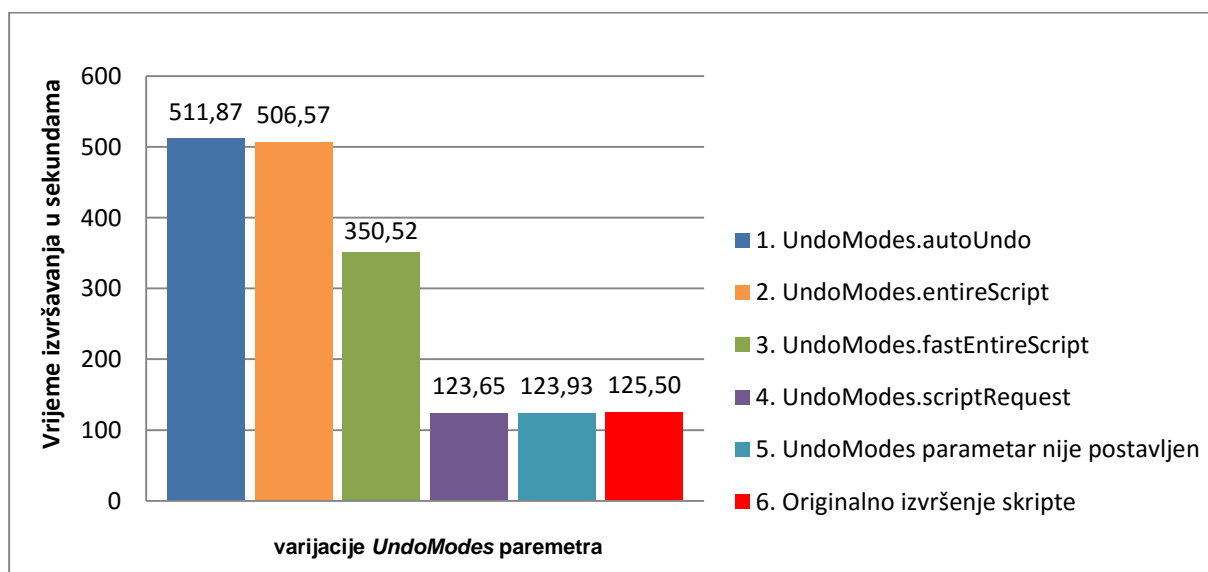
performansi nudeći dva parametra koji kontroliraju način kako se skripte izvršavaju relativno prema *Undo* opcijama InDesign programa.“ [29]

Korištene UndoModes opcije u Tablici 14. i na Slici 59. su:

1. *UndoModes.autoUndo* – ne dodaju se događaji u *Undo* redoslijed;
2. *UndoModes.entireScript* – dodaje se jedan događaj za cijelu skriptu u *Undo* redoslijed;
3. *UndoModes.fastEntireScript* – dodaje se jedan događaj za cijelu skriptu u *Undo* redoslijed;
4. *UndoModes.scriptRequest* – dodaje se svaka akcija skripte kao poseban događaj *Undo* redoslijed;
5. *UndoModes* parametar nije postavljen pri korištenju *doScript* metode;
6. Originalno izvršenje skripte bez *doScript* metode.

Tablica 14. Vrijeme izvršavanja skripte u sekundama mjereno pri variranju *UndoModes* parametra *doScript* metode

UndoModes opcije doScript metode	1	2	3	4	5	6
Vrijeme montaže	0,98	0,95	0,98	0,98	0,98	0,95
Vrijeme numeracije	457,04	459,98	300,32	73,39	73,81	73,46
Vrijeme čekanja	53,85	45,64	49,22	49,28	49,14	51,09
Ukupno vrijeme	511,87	506,57	350,52	123,65	123,93	125,50



Slika 59. Usporedba vremena izvršavanja skripte mjerena variranjem vrijednosti *UndoModes* parametra *doScript* metode

Iz Tablice 14. i Slike 59. može se zaključiti da tvrdnje u Adobe dokumentaciji [29] nisu istinite. Kad se koriste *UndoModes* parametri *doScript* metode, vrijeme izvođenja nije bilo moguće optimizirati. Umjesto optimizacije, vrijeme izvođenja povećano je 3 do 4 puta. U slučajevima korištenja **4.***UndoModes.scriptRequest* i **5.***UndoModes*, dobiveni rezultati gotovo su jednaki originalnom izvođenju skripte. U navedenim slučajevima skripta se izvršava na jednak način, zapisivanjem *Undo* koraka, jer je predefiniранo ponašanje *doScript* metode zapisivanje svih koraka.

5.4.1.3. Korištenje povezanih (eng. *linked*) umjesto samostalnih tekstualnih okvira

Originalno napisana skripta koristi odvojene tekstualne okvire u koje se skriptno ubacuje numeracija. Kako bi se odredilo utječe li korištenje povezanih okvira pri izradi pripreme na vrijeme izvršavanja skripte, originalno napisana skripta promijenjena je na način da koristi povezane okvire pri ubacivanju numeracije.

Kod mjerenja optimizacije upotrebljavane su unaprijed zadane postavke i varijacija je rađena na opciji *Broj polja za numeraciju po kartici* gdje je broj variran od jednog do pet okvira.

Proces numeracije započinje nakon izvršenja montaže na *Master* stranici dokumenta. Na *Master* stranici nalaze se montirane kartice (umetnuti PDF dokument) po kojima su pozicionirani tekstualni okviri unutar kojih će se na stranicama dokumenta ubacivati numeracija. Pri dodavanju potrebnih stranica u dokument montaža će se kopirati na nove stranice (jer je postavljena na *Master* stranici) pa su na novonastalim stranicama tekstualni okviri dostupni tako da se izvuku, tj. oslobode (eng. *override*) s *Master* stranice. Numeracija započinje dodavanjem potrebnog broja stranica u dokument što ovisi o montaži i zadanom rasponu numeracije.

Pri odabranoj postavki numeracije postavljene na povećavanje kroz arke (za izradu blokova sa slijednim brojevima) prvi će broj biti postavljen u tekstualni okvir prve kartice na prvoj stranici dokumenta, a drugi broj na drugu stranicu u isti okvir *Master* stranice. Originalna skripta koristi dvije ugniježdene *for* petlje. U ugniježdеноj *for* petlji nalazi se i *while* petlja (Slika 60.). U *while* petlji se na odabrani tekstualni okvir radi *override* i u okvir se ubacuje broj. *While* petlja se koristi kako bi se moglo ponoviti ubacivanje istog broja na više tekstualnih okvira na kartici (u testiranju je varijacija bila od 1 do 5 okvira). Unutarnja *for* petlja služi kako bi se kretali po stranicama pri ubacivanju slijednih brojeva. Vanjska *for*

petlja služi za prolaz kroz kartice. Okviri se ne povezuju između stranica pri ubacivanju brojeva numeracije.

```
function numeracijaBlok() {
  //dodavanje stranica
  myDocument.documentPreferences.pagesPerDocument = brojevaBlok + 1;
  brOkvira = 0;
  //kartica na sebi može imati jedan ili više tekstualnih okvira
  for (brK = 0; brK < brKartica; brK++) {
    //prolaz kroz stranice
    for (str = 1; str < brojevaBlok + 1; str++) {
      stranica = myDocument.pages.item(str);
      //broj ponavljanja se koristi kad je više tekstualnih okvira
      po stranici
      brPonavljanja = 0;
      while (brPonavljanja < brojPoljaZaNum) {
        masterMontaza.textFrames
        item(brOkvira + brPonavljanja).override(stranica);
        stranica.textFrames.item(brOkvira + brPonavljanja).contents=
        pad(numOD, leadZeroBr);
        brPonavljanja++;
      }
      numOD++;
    }
    brOkvira += brojPoljaZaNum;
  }
}
```

Slika 60. Originalno napisan dio koda koji ne koristi povezane priče

Promjene u originalnoj skripti:

Izmijenjena skripta koristi tri *for* petlje (Slika 61.). Vanjska *for* petlja služi za prolazak kroz okvire. U trenutku selekcije prvog okvira na kartici na prvoj stranici u dokumentu radi se *override* s *Master* stranice te se ugniježđenom *for* petljom izrađuju brojevi koji će ići kroz povezane okvire. Brojevi se izrađuju samo za prvi okvir na kartici. Ukoliko postoji više okvira na kartici, isti se sadržaj ubacuje i u njih. Nakon što su kreirani svi brojevi, oni se samo jednom ubacuju u okvir, a budući da će se okviri povezati kroz stranice, brojevi će se rasporediti kroz okvire koji se nalaze u istoj priči. Nakon navedenog koristi se dodatna *for* petlja za povezivanje tekstualnih okvira koji sadrže slijedne brojeve kroz dokument.

Tablica 15. i Slika 62. Prikazuju rezultate mjerenja vremena izvršavanja originalne i izmijenjene skripte. Iz rezultata prikazanih na Slici 62. i usporedbom vremena izvršavanja u tablicama 13. i 15. može se zaključiti da upotreba povezanih umjesto odvojenih okvira može skratiti vrijeme izvršavanja i optimizirati skriptu. Skripta koja koristi povezane okvire, u

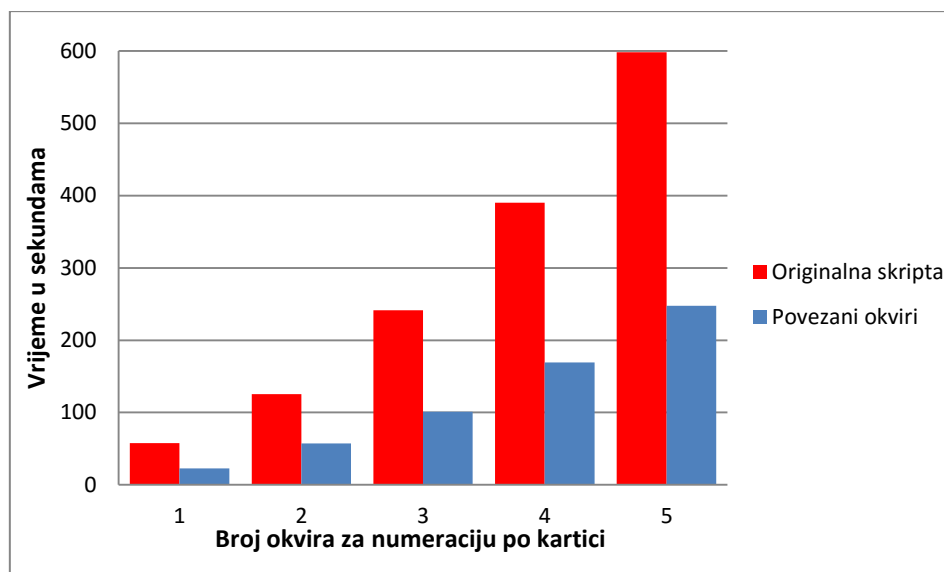
prosjeku je 2,4 puta brža od skripte koja koristi odvojene okvire. Na navedeni način moguće je upola smanjiti vrijeme izvršavanja i optimizirati skriptu. Iz usporedbe je vidljivo da se najviše uštedjelo na vremenu čekanja. Razlozi navedenog mogu biti u veličini datoteke koja u konačnici nastaje i u optimizaciji Adobe InDesigna za upotrebu povezanih okvira. Usporedba dobivenih veličina datoteka koje su izrađene sa i bez povezanih okvira, prikazana je u Tablici 16. i na Slici 63.

```
function numeracijaBlok() {
    //dodavanje stranica
    myDocument.documentPreferences.pagesPerDocument = brojevaBlok + 1;
    brOkviraNaM=masterMontaza.textFrames.length;
    brPonavljjanja=1;
    for (brOkvira=0; brOkvira<brOkviraNaM; brOkvira++) {
        if (brPonavljjanja==1) {
            //izdada tekstualne varijable koja će sadržavati sve brojeve
            koji se ubacuju u linkane okvire. Varijabla se izrađuje samo
            za prvi okvir na kartici a ubacuje ovisno o broju ponavljanja
            mojString="";
            for (i=0; i<brojevaBlok; i++) {
                mojString+=pad(numOD++, leadZeroBr) + '\r';
            }
        }
        stranica=myDocument.pages.item(1);
        stranica.masterPageItems[0].parent.textFrames.item(brOkvira).override(stranica);
        //ubacivanje teksta u prvi okvir kartice na prvoj stranici
        stranica.textFrames.item(brOkvira).contents=mojString;
        for (str=2; str<brojevaBlok+1; str++) {
            stranica=myDocument.pages.item(str-1);
            stranical=myDocument.pages.item(str);
            stranical.masterPageItems[0].parent.textFrames.item(brOkvira).override(stranical);
            //povezivanje okvira kroz stranice
            stranica.textFrames.item(brOkvira).nextTextFrame=
            stranical.textFrames.item(brOkvira);
        }
        if (brPonavljjanja<brojPoljaZaNum) brPonavljjanja++;
        else brPonavljjanja=1;
    }
}
```

Slika 61. Izmjene u kodu za upotrebu povezanih priča

Tablica 15. Vrijeme izvršavanja optimizirane skripte koja koristi povezane okvire u sekundama

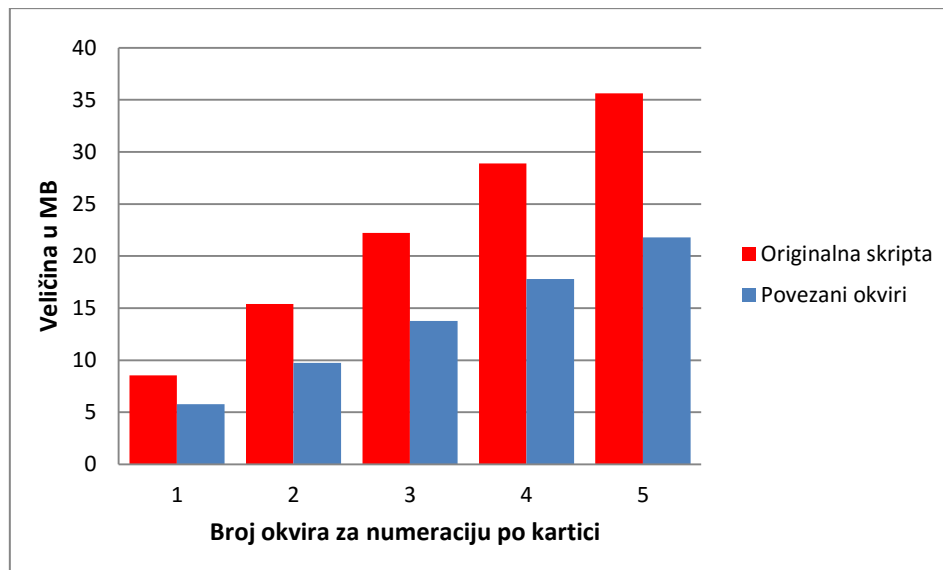
Broj okvira za numeraciju po kartici	1	2	3	4	5
Vrijeme montaže	0,84	0,98	1,13	1,30	1,40
Vrijeme numeracije	21,28	51,31	94,61	156,17	233,08
Vrijeme čekanja	1,23	5,13	5,12	11,64	13,08
Ukupno vrijeme	23,35	57,42	100,86	169,11	247,56



Slika 62. Usporedba vremena izvršavanja originalne skripte i skripte koja koristi povezane okvire

Tablica 16. Veličine dobivenih InDesign datoteka (.indd) u MB originalne skripte i optimizirane skripte koja koristi povezane okvire

Broj okvira za numeraciju po kartici	1	2	3	4	5
Originalna skripta	8,53	15,40	22,21	28,91	35,62
Optimizirana skripta koja koristi povezane okvire	5,76	9,75	13,76	17,79	21,79



Slika 63. Usporedba veličina InDesign datoteka u MB originalne skripte i skripte koja koristi povezane okvire

5.4.1.4. Funkcija „pad“

Pri izvođenju skripte uzastopce poziva se funkcija „pad“. Uloga funkcije je dodavanje nula (eng. *leading zero*) ispred broja numeracije koji se ubacuje. Time se želi postići da su svi brojevi jednake duljine. Kako se funkcija poziva velik broj puta pri izvođenju (za svaki broj koji se ubacuje u tekstualni okvir), želio se odrediti njezin utjecaj na konačno vrijeme izvođenja. U originalnoj skripti upotrijebljena je funkcija „pad1“. Ostale testirane funkcije mogu se vidjeti na Slici 64.

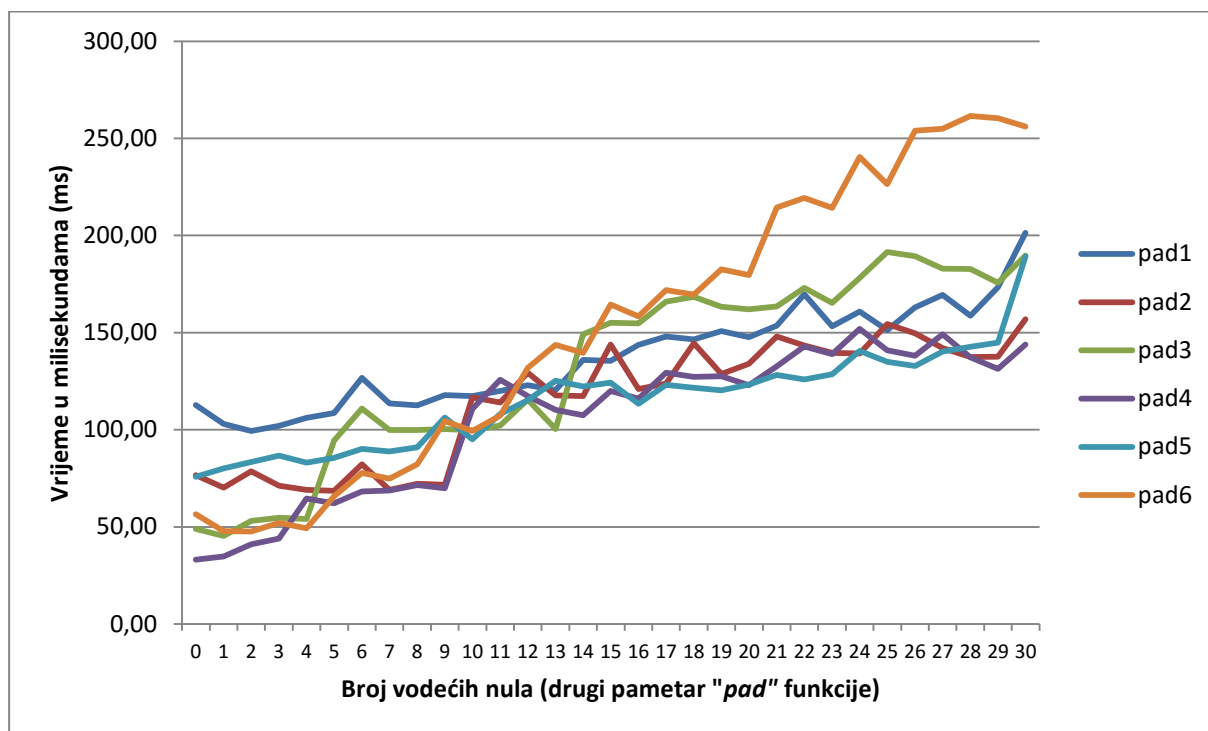
Napisana je posebna skripta koja mjeri vrijeme pojedine funkcije. Funkcijama su se prosljeđivali brojevi 1-10.000 kao prvi parametar. Drugi parametar je bio broj vodećih nula u rasponu od 0 do 30. Dakle, testirala se numeracija do 10.000, a broj vodećih nula varirao je u prolazima od nula do 30 znamenki. Slike 65. i 66. prikazuju dobivene rezultate.

```

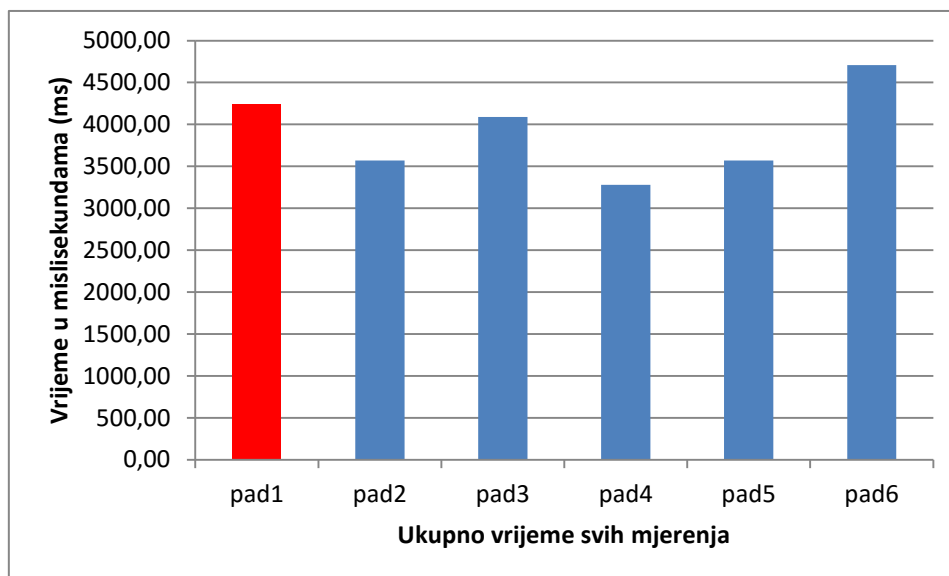
function pad1(number, digits) {
  return Array(Math.max(digits - String(number).length + 1, 0)).join(0) + number;
}
function pad2(num, size) {
  return (Math.pow(10, size) + ~ ~num).toString().substring(1);
}
function pad3(n, length) {
  var s = n + "", needed = length - s.length;
  if (needed > 0) s = (Math.pow(10, needed) + "").slice(1) + s; return s;
}
function pad4(number, pad) {
  var N = Math.pow(10, pad);
  return number < N ? (" " + (N + number)).slice(1) : "" + number;
}
function pad5(n, len) {
  return (new Array(len + 1).join('0') + n).slice(-len);
}
function pad6(number, length) {
  var my_string = "" + number;
  for (var to_add = length - my_string.length; to_add > 0; to_add -= 1) {
    my_string = '0' + my_string;
  }
  return my_string;
}

```

Slika 64. Testirane funkcije za dodavanje nula ispred broja



Slika 65. Prikaz izmjerenih vremena „pad“ funkcija



Slika 66. Ukupno vrijeme pojedine „pad“ funkcije u svim testovima

Iz grafikona je vidljivo da je bila upotrijebljena druga najsporija funkcija „pad1“. Najbrža od navedenih funkcija je „pad4“, ali kod nje se, kao i kod funkcija „pad2“ i „pad3“, javio problem prilikom ispisa brojeva u programu jer su brojevi bili prikazani kao eksponent (pr. Ispisa „e+ 27“). Funkcije „pad4“ i „pad2“ ne prikazuju dobro brojeve s dodanim nulama nakon brojeva s 15 pozicija, dok funkcija „pad3“ ne prikazuje željene rezultate nakon brojeva s 22 pozicije. Razlog tomu je korištenje JavaScript `Math.pow()` funkcije koja vraća eksponent.

Budući da je funkcija „pad4“ najbrža, ona je odabrana za rad s dodavanjem nula za raspon od 2 do 15 pozicija brojeva s vodećim nulama, a za ostala dodavanja funkcija „pad5“ kao sljedeća najbrža funkcija koja prikazuje dobar rezultat.

Prije novog testiranja skripta je izmijenjena tako da nije potrebno pri pozivanju funkcije prosljeđivati broj pozicija s vodećim nulama u broju, već se broj znamenki ugrađuje pri dinamičkoj definiciji funkcije kako je prikazano na Slici 67.

```

if (leadZeroBr < 2) {
  pad = function (broj) {
    return broj+"";
  }
} else if (leadZeroBr < 16) {
  var N = Math.pow(10, leadZeroBr);
  pad = function (broj) {
    return broj < N ? ("".padStart(leadZeroBr, '0') + broj).slice(1) : "" + broj;
  }
} else {
  pad = function (broj) {
    return (new Array(leadZeroBr + 1).join('0') + broj).slice(-leadZeroBr);
  }
}
}

```

Slika 67. Izmijenjena pad funkcija

Tablica 17. Rezultati testiranja skripte s optimiziranom pad funkcijom u sekundama

Broj tekstualnih okvira i raspon numeracije	1 tekstualni okvir - 1000 brojeva				2 tekstualna okvira - 1000 brojeva			
	pad 0	pad 5	pad 10	pad 20	pad 0	pad 5	pad 10	pad 20
Varijacije u broju vodećih nula								
Originalna skripta - vrijeme numeracije	30,75	30,68	30,96	30,81	73,926	74,31	74,44	74,159
Skripta s izmijenjenom pad funkcijom - vrijeme numeracije	30,347	29,97	30,72	30,43	73,372	73,98	73,468	73,374
RAZLIKA	0,4033	0,71	0,24	0,38	0,554	0,33	0,972	0,785

Tablica 17. prikazuje prosječno vrijeme numeracije na 5 mjerenja. Prikazano je samo vrijeme numeracije, a ne izvršavanje cijele skripte. Postavke pri mjerenjima bile su jednake početno zadanim parametrima izvođenja skripte, osim varijacije vodećih nula koje nisu rađene u drugim testovima. Mjerenja su rađena na raspon numeracije od 1 do 1000 i variranjem broja okvira po kartici na 1 ili 2 okvira. Iz dobivenih rezultata može se zaključiti da izmijenjena „pad“ funkcija nije imala značajan utjecaj na vrijeme izvođenja skripte. Može se primijetiti da povećavanjem broja nula gotovo da i nema razlike u izvođenju numeracije.

5.4.1.5. Korištenje *everyItem()* za dohvaćanje podataka u objektima kolekcije (eng. *Collection objects*)

Objekti u kolekciji su objekti koji dijele jednaka svojstva i čine grupu objekata. Pojedinačne objekte iz kolekcije objekata moguće je dohvaćati upotrebom indeksa. Jedan od načina dohvaćanja svih objekata u kolekciji je upotreba *everyItem()* funkcije [59]. U testovima se željelo istražiti kako upotreba funkcije *everyItem* za mijenjanje svojstava svih objekata u kolekciji utječe na vrijeme izvršavanja skripte u usporedbi s početnim načinom mijenjanja svojstava originalne skripte gdje se ona mijenjaju pojedinačnim dohvaćanjem objekata.

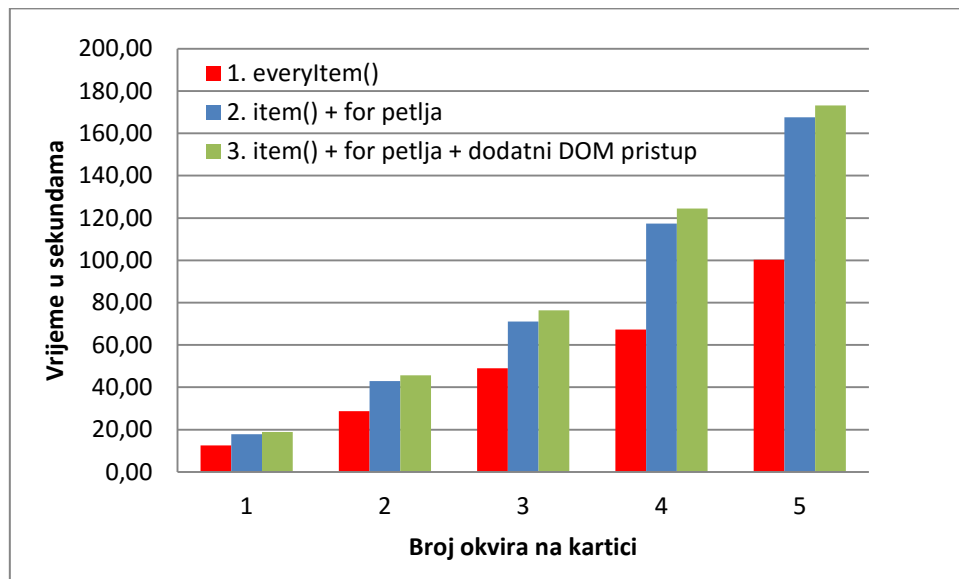
Testiranje vremena izvršavanja skripti rađeno je na unaprijed pripremljenim dokumentima od 50 stranica na kojima je bila postavljena *Master* stranica s okvirima kao i u ostalim testiranjima. Testiralo se samo vrijeme otključavanja okvira kao što se može vidjeti iz primjera koda na Slici 68. Testiranja su rađena na dokumentima u kojima se na montaži nalazi 1 do 5 okvira po montiranoj kartici.

```
//1. everyItem()
for (str = 0; str < 50; str++) {
    stranica = myDocument.pages.item(str);
    masterMontaza.textFrames.everyItem().override(stranica);
}
//2. item() i for petlja
for (str = 0; str < 50; str++) {
    stranica = myDocument.pages.item(str);
    for (i = 0, broj= masterMontaza.textFrames.length; i < broj; i++) {
        masterMontaza.textFrames.item(i).override(stranica);
    }
}
//3. item() i for petlja + dodatni DOM pristup
for (str = 0; str < 50; str++) {
    stranica = myDocument.pages.item(str);
    for (i = 0; i < masterMontaza.textFrames.length; i++) {
        masterMontaza.textFrames.item(i).override(stranica);
    }
}
```

Slika 68. Skripte korištene za testiranje rada s objektom kolekcije

Iz dobivenih vremena skripte može se zaključiti da ukoliko je potrebno mijenjati svojstva svih elementa u kolekciji, brža je uporaba *everyItem()* u usporedbi s mijenjanjem svojstava na zasebnim objektima. Iz grafikona prikazanog na Slici 69. može se primijetiti da je razlika u vremenu izvršavanja skripte veća ukoliko radimo s više objekata u kolekciji. Također se može primijetiti da i dodatni pristup DOM-u utječe na vrijeme izvršavanja. Ukoliko se smanji broj pristupa DOM-u, smanjuje se i vrijeme izvršavanja skripte. Smanjenje

broja pristupa DOM-u moguće je upotrebom varijabli u koje se spremaju reference prema DOM-u (Slika 68. primjeri testiranih kodova).



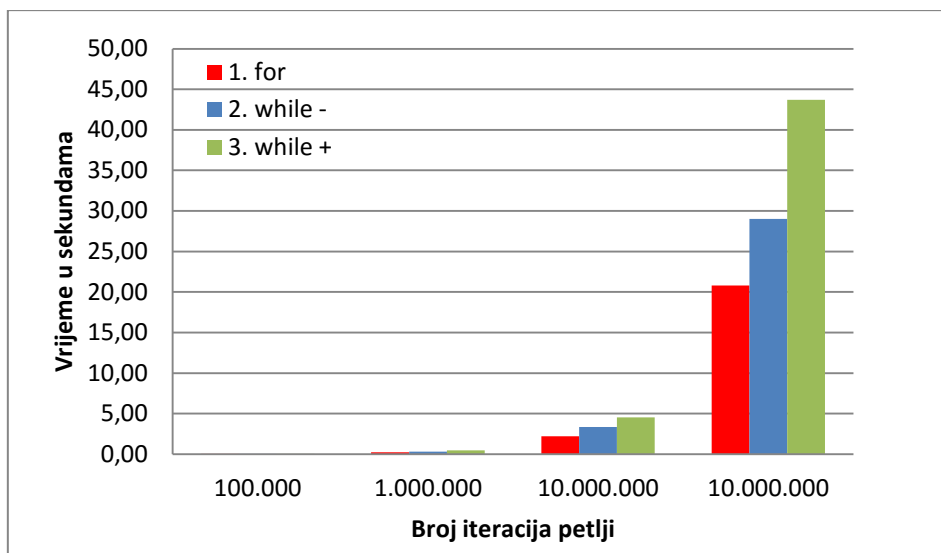
Slika 69. Usporedba vremena rada s kolekcijom i pojedinačnim pristupom objektima te prikaz razlike u vremenu izvođenja kad se smanji broj pristupa DOM-u

5.4.1.6. Testiranje brzine izvršavanja *for* i *while* petlji

Za testiranje brzine *for* i *while* petlji izrađena je posebna skripta što se može vidjeti na Slici 70. Mjerenja su rađena na 100.000, 1.000.000, 10.000.000 i 100.000.000 iteracija petlji.

```
//varijabla broj se postavlja u iteracijama na vrijednosti:  
//100.000, 1.000.000, 10.000.000, 100.000.000  
a = b = c = br2 = 0; br1 = broj;  
for (br = 0; br < broj; br++) {a++;} //1. for  
while (br1--) {b++;} //2. while -  
while (br2 < broj) {c++; br2++;} //3. while +
```

Slika 70. Skripte upotrijebljene za testiranje *for* i *while* petlje



Slika 71. Usporedba brzine izvršavanja *for* i *while* petlji

Iz dobivenih rezultata može se zaključiti da izvršavanje *for* petlje traje kraće nego kod *while* petlje. Vrijeme izvršavanja *while* petlje ovisi o tome izvršava li se petlja smanjivanjem uvjeta ili povećavanjem. *While* petlja će vremenski trajati kraće ukoliko se izvršava smanjivanjem uvjeta. Navedena optimizacija je optimizacija JavaScript koda i nema izravnog doticaja s DOM-om dokumenta. Kako se može vidjeti iz Slike 71., uštede u vremenu su relativno malene i primijenjene na skriptu koja se optimizira donose vrlo male uštede u vremenu izvršavanja. U posljednjem je testu na 100.000.000 iteracija petlje došlo do značajnije uštede u vremenu, ali navedena situacija nije toliko česta u izvršavanju zadataka skriptiranjem. Iz navedenog se može zaključiti da optimiziranje funkcija ili zadataka koji ne pristupaju DOM-u ne dovodi do značajnijeg ubrzanja skripte koja izvršava tipične zadatke grafičke pripreme.

5.4.1.7. Konačna optimizacija skripte

Kako bi se skripta pripremila za upotrebu, primijenjene su dobivene spoznaje i ponovno je izrađena optimizacija na skriptni kod sa Slike 61. što je prikazano na Slici 72.

Na Slici 72. možemo vidjeti optimizacije primijenjene na konačnu skriptu. Primijenjeno je više optimizacija kako slijedi:

1. Upravljanje dosegom varijabli:

Kad se JavaScript kod izvršava, kreira se kontekst izvođenja, tj. doseg varijabli (eng. *execution context*, *variable scope*). Kontekst izvođenja određuje

okruženje u kojemu se kod izvršava. Globalni kontekst izvođenja kreira se početkom izvršavanja skripte, a dodatni konteksti pri izvršavanju funkcija. U konačnici postoji stog (eng. *stack*) sačinjen od konteksta izvršavanja gdje je gornji kontekst onaj koji je trenutno aktivan. Performanse razrješenja referenci izravno ovise o broju objekata koji se pretražuju u lancu dosega. Kako se varijable nalaze dalje u stogu, dulje će trajati potraga, tj. vrijeme potrebno za pristupanje traženoj varijabli je dulje. Ukoliko se objektima pristupa neispravno, s kontekstom izvođenja koji ima dug lanac dosega, može se dobiti negativan efekt na vrijeme izvršavanja [60] [61].

Iz navedenih razloga sve su varijable koje se koriste u funkcijama u skripti postavljene kao lokalne što poboljšava vrijeme izvođenja skripte zbog smanjenog konteksta izvođenja i lanca dosega pri razrješenju varijabli.

```
function numeracijaBlok() {
  myDocument.documentPreferences.pagesPerDocument = brojevaBlok + 1;
  //postavljanje svih varijabli na lokalne varijable kako bi se smanjilo kontekst
  //izvođenja i lanac dosega varijabli
  var brOkvira, mojString, stranica, stranica1, str, brOkvKartica = brojPoljaZaNum,
  brojStr = brojevaBlok;
  //spremanje u varijable i ponovno korištenje riješenih referenci
  var broj = masterMontaza.textFrames.length;
  var okvMaster = masterMontaza.textFrames.everyItem();
  var strDoc = myDocument.pages;

  okvMaster.override(strDoc.item(1));
  stranica = strDoc.item(1).textFrames.everyItem().getElements();
  var brPonavljanja = 1;
  for (brOkvira = 0; brOkvira < broj; brOkvira++) {
    if (brPonavljanja == 1) {
      mojString = "";
      for (i = 0; i < brojStr; i++) {
        mojString += pad(numOD++) + '\r';
      }
      mojString = mojString.slice(0, -1);
    }
    stranica[brOkvira].contents = mojString;
    if (brPonavljanja < brOkvKartica) brPonavljanja++;
    else brPonavljanja = 1;
  }
  for (str = 2; str <= brojStr; str++) {
    okvMaster.override(strDoc.item(str));
    stranica1 = strDoc.item(str).textFrames.everyItem().getElements();
    stranica = strDoc.item(str - 1).textFrames.everyItem().getElements();
    for (brOkvira = 0; brOkvira < broj; brOkvira++) {
      stranica[brOkvira].nextTextFrame = stranica1[brOkvira];
    }
  }
}
```

Slika 72. Finalno optimizirana skripta

2. Rad s objektima kolekcije:

JavaScript smatra *everyItem()* jedinstvenim objektom iako bilo koje svojstvo ili metoda koji se pozivaju nad objektom kolekcije šalje naredbu višestrukim objektima koji će je primiti. Ovakav mehanizam poznat je kao *verb-first command*; umjesto pozivanja specificirane metode na svakom objektu, jedna će metoda odraditi akciju na većem broju objekata [62].

Kako je dokazano pod 5.4.1.5., izvođenje zadataka nad objektima kolekcije pomaže u bržem izvršavanju skriptnih zadataka pa se kod izmijenio tako da koristi objekte kolekcije umjesto odvojenog izvršavanja zadataka na pojedinačnim objektima.

3. Spremanje razriješenih referenci prema DOM-u u varijable:

U finalnoj skripti izrađena je i optimizacija spremanja razriješenih referenci prema DOM-u, koje će se više puta koristiti, u lokalne varijable. Takvim se pristupom iste reference ne trebaju na više mjesta razrješavati i smanjuje se broj pristupa DOM-u [62].

Kako bi se razriješile reference prema objektima kolekcije, najboljom se opcijom pokazala funkcija *getElements()* koja razriješene reference sprema u polje. Prema navodima pronađenim u dostupnoj literaturi [59], ističe se da je pri radu s objektima kolekcije bolje raditi s poljima, tj. pretvarati odabranu kolekciju u polje te zatim nad poljem izvršavati potrebne zadatke. Iako se objektima kolekcije pristupa na isti način kao i poljima, korištenjem indeksa, ističe se da je mnogo brže procesirati polja u usporedbi s kolekcijama.

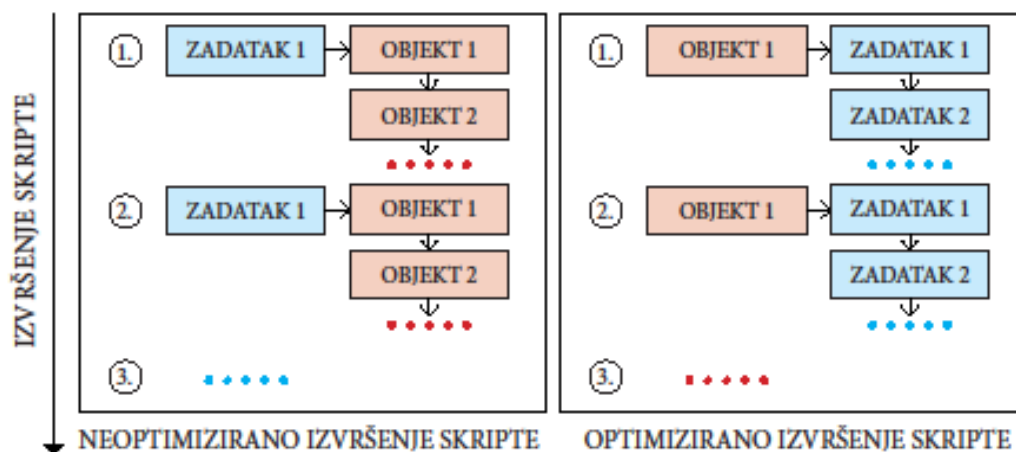
4. Upotreba samo *for* petlji koje se optimiziraju kako bi se smanjio pristup DOM-u dokumenta:

U kodu se upotrebljavaju samo *for* petlje zbog rezultata testiranja u poglavlju 5.4.1.6. čiji su uvjeti optimizirani da pri provjeri ponovne iteracije ne pristupaju DOM-u.

5. Izrada skriptnih zadataka primjenom lokalnosti u pristupanju DOM-u:

Ranije nabrojene optimizacije donose manju vremensku uštedu, tek u nekoliko sekunda. Najznačajnija je optimizacija, koja donosi najveću uštedu u vremenu, spajanje dvije *for* petlje koje prolaze kroz stranice na Slici 61. u jednu *for* petlju (Slika 72.). Prva *for* petlja služila je za otključavanje okvira, a druga za povezivanje okvira po stranicama. Kad su te dvije petlje spojene u jednu, dobivena je značajna ušteda u vremenu.

Iz navedenog može se zaključiti da kad se skripta optimizira na način da se svi zadatci odrađuju na jednom selektirani objekt iz DOM-a, dobivaju se velike uštede u vremenu izvođenja skripte za razliku od odrađivanja zadataka na način da se rješava zadatak po zadatak na svim objektima kako prikazuje Slika 73. Pri radu s DOM objektima najbolje je upotrebljavati vremensku i prostornu lokalnost odrađivanjem svih potrebnih zadataka u istom prolazu.



Slika 73. Optimizacija izvršavanja skripte upotrebom lokalnosti u izvršavanju zadataka

Iz dobivenih rezultata prikazanim na Slici 74. i tablicama 18. i 19. možemo zaključiti da optimizacijom ExtedndScript koda možemo dobiti značajne uštede u vremenu izvršavanja skripte, osobito kod dugotrajnih skripti. Optimizacijom se vrijeme izvršavanja smanjilo 3 - 4 puta. Tako se primjerice vrijeme izvršavanja skripte kod 5 okvira na kartici, tj. 5.000 okvira u dokumentu, smanjilo od početnih 10-ak minuta na oko 2,5 minute. Dobivena ušteda je značajna te je vrijeme izvršavanja skripte skraćeno za 76 %.

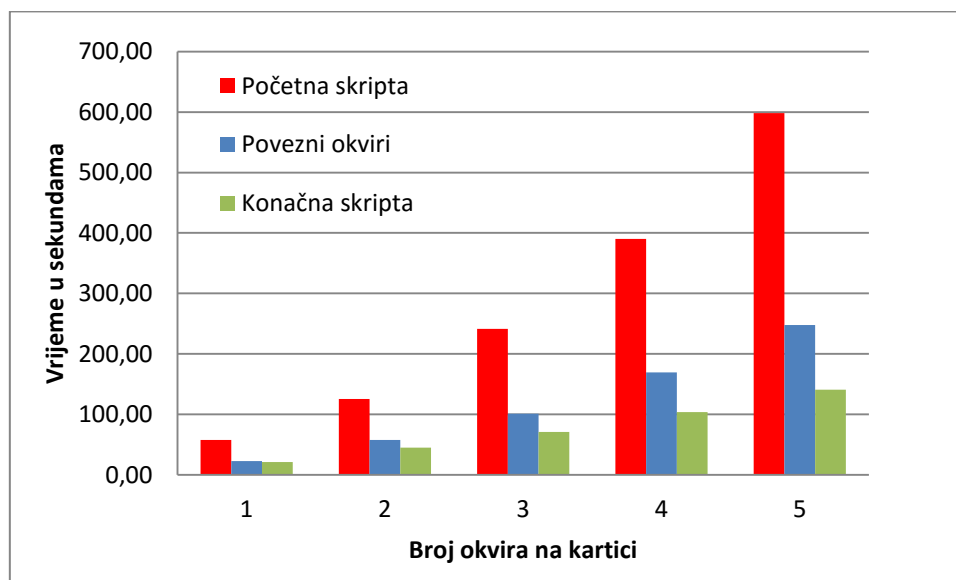
Ukupno vrijeme za sva mjerenja smanjeno je prosječno 70 %. Optimizirana će se skripta u prosjeku izvršavati u 30 % vremena početno napisane skripte.

Tablica 18. Rezultati mjerenja konačne skripte

Broj okvira za numeraciju po kartici	1	2	3	4	5
Vrijeme montaže	0,83	0,98	1,11	1,25	1,38
Vrijeme numeracije	19,16	42,25	67,01	96,42	130,84
Vrijeme čekanja	1,11	1,77	2,63	6,16	8,51
Ukupno vrijeme	21,10	45,00	70,75	103,83	140,73

Tablica 19. Ušteda vremena iskazana u postotcima između početne i konačne skripte

Broj okvira za numeraciju po kartici	1	2	3	4	5
Ušteda vremena u postotcima konačne skripte prema početnoj	64 %	64 %	71 %	73 %	76 %



Slika 74. Usporedba početne skripte, skripte s povezanim okvirima i konačne skripte

5.4.2. Rezultati usporedbe standardnog i skriptnog načina izvođenja zadatka

Testiranje obavljanja zadataka izrade varijabilne grafičke pripreme, koja uključuje izradu numeracije i montaže kartica, rađeno je na skriptni i standardni način. Obavljanje zadataka na standardni i skriptni način opisano je u poglavlju 4.3.4., a dobiveni rezultati pojedinačnih mjerenja u poglavlju Prilozi u Tablici 46.

Pri testiranju su upotrebljavane postavke kao i u ranijim testovima s varijacijom opcije *Broj polja za numeraciju po kartici* od 1 do 5. Tablice 20. i 21. te slike 75. i 76. prikazuju dobivene rezultate.

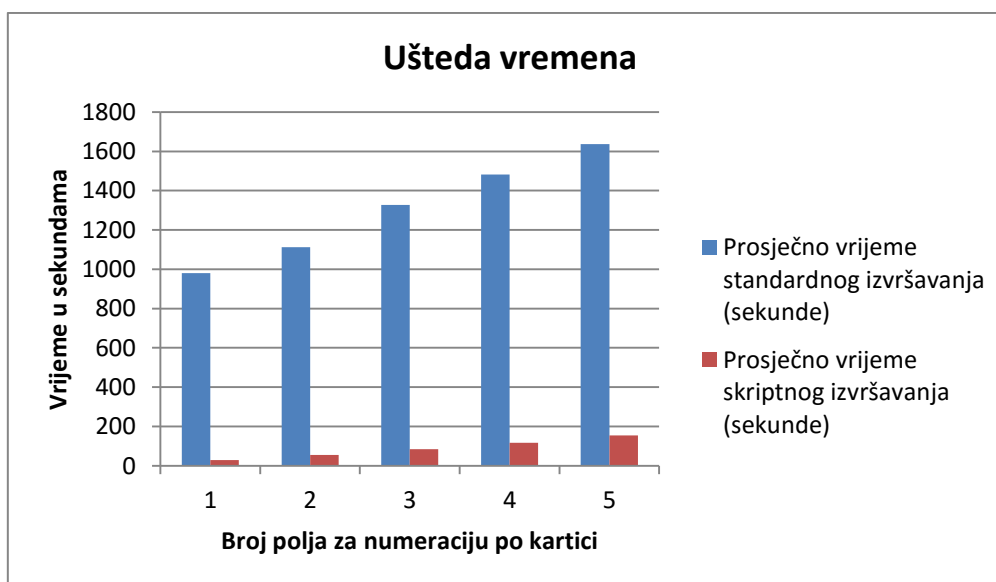
Dobiveni rezultati prikazuju prosječno 96 % smanjenja korisničkog unosa te visok dobiveni nivo automatizacije. Vrijeme je također uvelike smanjeno, u prosjeku za 94 %. Time je dokazana hipoteza da analizom grafičke pripreme možemo detektirati poslove koje je poželjno automatizirati skriptiranjem. Poslovi koji zahtijevaju mnogo ručnih zadataka i koji su ponavljajući, mogu se optimalno automatizirati uz pomoć skriptiranja.

Tablica 20. Usporedbe vremena skriptnog i standardnog izvođenja zadatka numeracije i montaže kartica

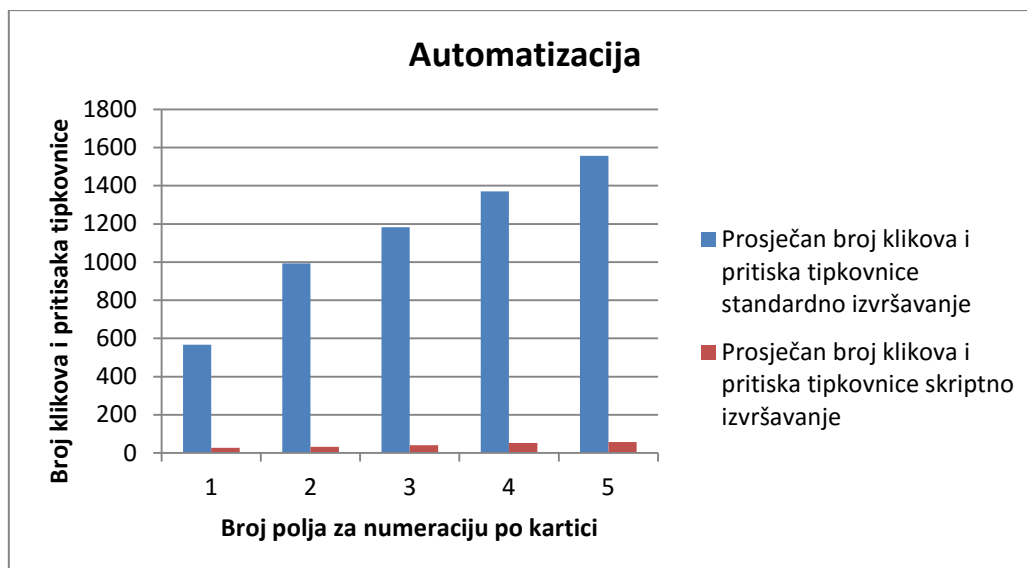
Broj polja za numeraciju po kartici	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postotci)	Faktor smanjenja
1	980	28	952	97 %	1:35
2	1113	55	1058	95 %	1:20
3	1327	84	1243	94 %	1:16
4	1482	116	1366	92 %	1:13
5	1637	154	1483	91 %	1:11

Tablica 21. Usporedbe broja klikova i pritisaka tipkovnice skriptnog i standardnog izvođenja zadatka numeracije i montaže kartica

Broj polja za numeraciju po kartici	Prosječan broj klikova i pritisaka tipkovnice standardno izvršavanje	Prosječan broj klikova i pritisaka tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
1	568	27	95 %
2	994	33	97 %
3	1182	41	97 %
4	1370	52	96 %
5	1556	58	96 %



Slika 75. Prikaz uštede vremena skriptnim izvođenjem zadatka numeracije i montaže kartica



Slika 76. Prikaz dobivene automatizacije skriptnim u usporedbi sa standardnim izvođenjem zadatka numeracije i montaže kartica

Tablica 22. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka numeracije i montaže kartica

Broj polja za numeraciju po kartici	1		2		3		4		5		Ukupno/prosjek	
	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno
Minimalan broj (b_{min})	552	24	968	28	1101	39	1326	47	1523	55	-	-
Maksimalan broj (b_{max})	590	29	1023	35	1256	43	1402	55	1586	63	-	-
Raspon varijacije (RV)	38	5	55	7	155	4	76	8	63	8	-	-
Prosjek (\bar{b})	568	27	994	33	1182	41	1370	52	1556	58	1134	42
Standardna devijacija (σ)	16	2	24	3	62	2	30	4	27	4	-	-
Koeficijent varijacije (V)	3 %	7 %	2 %	9 %	5 %	4 %	2 %	7 %	2 %	6 %	3 %	6 %
Najveće odstupanje od minimalne vrijednosti (V_{max})	7 %	21 %	6 %	25 %	14 %	10 %	6 %	17 %	4 %	15 %	14 %	25 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	3 %	11 %	3 %	16 %	7 %	6 %	3 %	10 %	2 %	5 %	4 %	10 %

Kod izračuna vjerojatnosti da će se zadatak ostvariti s minimalnim brojem koraka kod standardnog rješavanja zadatka može se zaključiti da je vjerojatnost minimalnog opterećenja operatera, tj. vjerojatnost da pri rješavanju zadatka neće doći do pogrešaka u izvođenju, jednaka nuli, a stopa pogrešaka iznimno velika. Tablica 22. prikazuje varijacije u izvođenju zadatka iz kojih možemo izračunati vjerojatnosti za standardni i skriptni način izvođenja zadatka:

$$P(\text{standardno}) = (1 - 0,04)^{1134} = 0 \%$$

$$P(\text{skriptno}) = (1 - 0,10)^{42} = 1,2 \%$$

Iz mjerenja je vidljivo da je vjerojatnost ostvarenja zadatka s minimalnim brojem koraka kod skriptnog izvođenja također niska jer pri izradi zadatka sudjeluje korisnik koji različito uspješno može ostvarivati zadatke koje je potrebno obaviti. Najviše se varijacija pri izradi zadatka skriptiranjem u broju klikova i pritisaka tipki tipkovnice nalazi pri pozicioniranju okvira na kartici te kod unošenja podataka u dijaloške okvire. Ostali dijelovi izvršavanja zadataka variraju vrlo malo jer skripta samostalno odrađuje zadatke za koje je inače zadužen korisnik. Iz navedenog se može zaključiti da je pri skriptnoj izradi vjerojatnost da neće doći do pogrešaka u postupku izrade, pa i u samom dokumentu, ovisna o korisniku. Za skripte koje ne uključuju interakciju s korisnikom, vjerojatnost minimalnog broja koraka je osigurana. Prednost skripti koje uključuju selekciju korisnika je u tome što se mogu primijeniti na više različitih zadataka s varijabilnim parametrima koji ovise o selekciji.

Pri pregledu gotovih dokumenata nisu pronađene pogreške u prijelomu. Sve pogreške pri standardnom izvođenju zadatka otklonjene su kako su nastajale.

5.5. Rezultati mjerenja skripte za izradu montaže i personalizacije kartica iz podataka u XML datoteci (S5)

Skripta za izradu montaže i personalizacije kartica iz podataka u XML datoteci izrađena je kako bi se testirala mogućnost optimizacije rada s XML datotekama u programu Adobe InDesign primjenom skriptnih tehnologija.

Tijek izvršavanja skripte i njezine mogućnosti opisane su u poglavlju 4.3.5. gdje je opisan i standardni način izvršavanja zadatka. Rađeno je po 5 mjerenja na 3 različita XML dokumenta na standardan i skriptni način. XML dokumenti odabrani su tako da se razlikuju po broju elemenata koje je potrebno pozicionirati po karticama. Osim razlike u broju elemenata po karticama, grafičke se pripreme razlikuju i u konačnom broju personaliziranih kartica koje se dobivaju jer XML dokumenti variraju i u dužini.

Izrada grafičke pripreme koja se testirala uključuje i izradu dizajna. Kako pripremanje tekstualnih okvira i stilova teksta može iznimno varirati i u skriptnom i u standardnom prijelomu, kod testiranja nije mjereno vrijeme i potreban broj klikova mišem i pritisaka tipki tipkovnice za pozicioniranje okvira na kartici i postavljanje oblikovanja teksta. Za sva testiranja izrađen je predložak na kojemu se nalaze postavljeni tekstualni okviri s oblikovanim tekstom. Tako se osiguralo da se u konačnici za sva testiranja dobiju identične pripreme i iz usporedbe izbace varijacije koje nastaju pri izradi dizajna.

Kod testiranja standardnog izvršavanja zadatka uključeni su sljedeći koraci:

1. Izrada novog dokumenta u programu Adobe InDesign veličine konačnog tiskovnog arka
2. Ubacivanje ranije izrađenog predloška u nastali dokument
3. Izrada reznih oznaka uz karticu te montiranje kartica
4. Povezivanje tekstualnih okvira kako bi se moglo automatizirati ubacivanje XML elemenata
5. Kreiranje stilova iz formatiranog teksta
6. Uvoženje XML oznaka u *Tags* paletu te povezivanje kreiranih stilova i uvezenih oznaka (opcija: *Tags/Map Tags to Styles*)
7. Zadavanje XML strukture u *Structure* paleti
8. Uvoženje XML-a i *Autoflow* sadržaja.

Kod testiranja izrađene skripte skriptno izvršavanje zadatka razlikuje se od opisa u poglavlju 4.3.5. jer se kod testiranja nije zadavao broj polja za XML nego su tekstualna polja s oblikovanim tekstom bila kopirana i ubačena iz predloška za skriptno izvršavanje zadatka.

Kod testiranja skriptne i standardne izrade zadatka izrađivani su zadatci prema sljedećim postavkama:

1. Izrada akreditacija (Slika 77.)

Format: 63 x 93 mm

Montaža: arak 483 x 330 mm, 18 kartica na arku, 5 araka

Napusti: 3 mm

XML: 4 XML elementa, 83 kartice

2. Izrada dopisnica (Slika 78.)

Format: 150 x 100 mm

Montaža: arak 483 x 330 mm, 9 kartica na arku, 49 araka

Napusti: 3 mm

XML: 3 XML elementa, 440 kartica

3. Izrada kodova (Slika 79.)

Format: 85 x 54 mm

Montaža: arak 483 x 330 mm, 25 kartica na arku, 97 araka

Napusti: 3 mm

XML: 2 XML elementa, 2420 kartica



Slika 77. Izrada personaliziranih akreditacija iz XML dokumenta (lijevo: predložak, desno gotova priprema)



Slika 78. Izrada personaliziranih dopisnic iz XML dokumenta (lijevo: predložak, desno gotova priprema)



Slika 79. Izrada personaliziranih kodova iz XML dokumenta (lijevo: predložak, desno gotova priprema)

Dobiveni rezultati pojedinačnih mjerenja prikazani su u poglavlju Prilozi u Tablici 47. Tablice 23. i 24. te slike 80. i 81. prikazuju prosječne rezultate za sva vršena mjerenja. Iz dobivenih rezultata vidljivo je da se korištenjem skriptnih tehnologija može ubrzati i olakšati rad s XML podatcima pri izradi grafičke pripreme.

Rezultati prikazuju da se vremenska ušteda u postotcima smanjuje povećanjem veličine XML dokumenta. Kod velikih dokumenata udio vremena rada korisnika kod standardnog i skriptnog izvršavanja je manji. Kod standardnog izvršavanja dio zadatka se također ostvaruje automatski - kod *Autoflow* postavljanja sadržaja iz XML elementa. Tako

će i u jednom i u drugom slučaju zadatak ubacivanja teksta biti ostvaren automatski. Na to vrijeme korisnik nema utjecaja pa se vremenska razlika smanjuje. Najveća se ušteda u vremenu kod skriptnog izvršavanja zadatka postiže kod pripreme dokumenta za prijelom (postavljanje reznih oznaka, montiranje, izrada stilova teksta, postavljanje strukture prijeloma...). Kako je udio vremena pripreme dokumenta u ukupnom vremenu izvršenja zadatka manji, tako se smanjuje i dobivena vremenska ušteda. Iz navedenog možemo zaključiti da će dobivena vremenska ušteda ovisiti o veličini XML dokumenta koji se prelama, montaži koju je potrebno napraviti i broju elemenata koji se pozicioniraju po kartici.

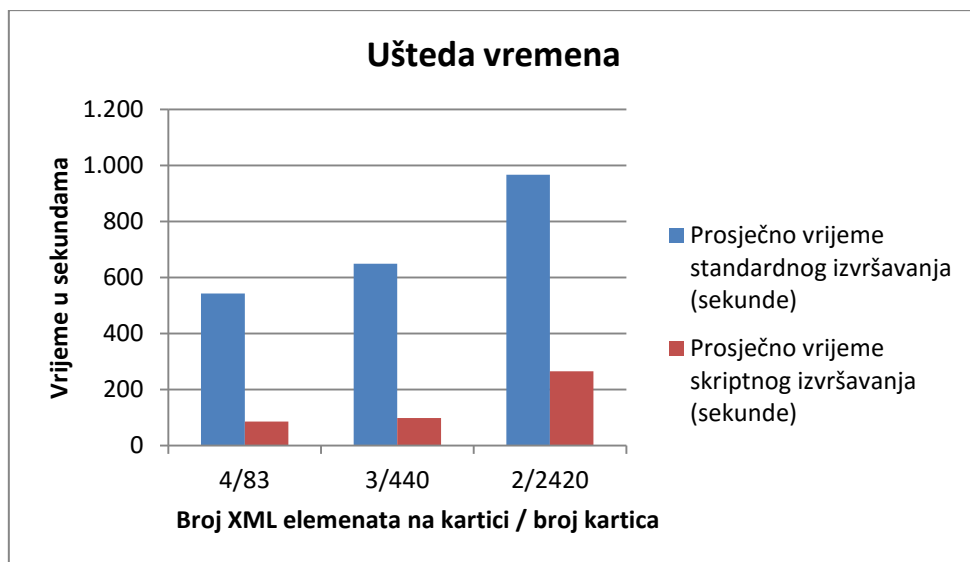
Tablica 23. Usporedbe vremena skriptnog i standardnog izvođenja zadatka izrade montaže i personalizacije kartica iz podataka u XML datoteci

Zadatak	Broj XML elemenata na kartici / broj kartica	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postotci)
1.	4/83	543	85	457	84 %
2.	3/440	649	98	551	85 %
3.	2/2420	967	265	702	73 %

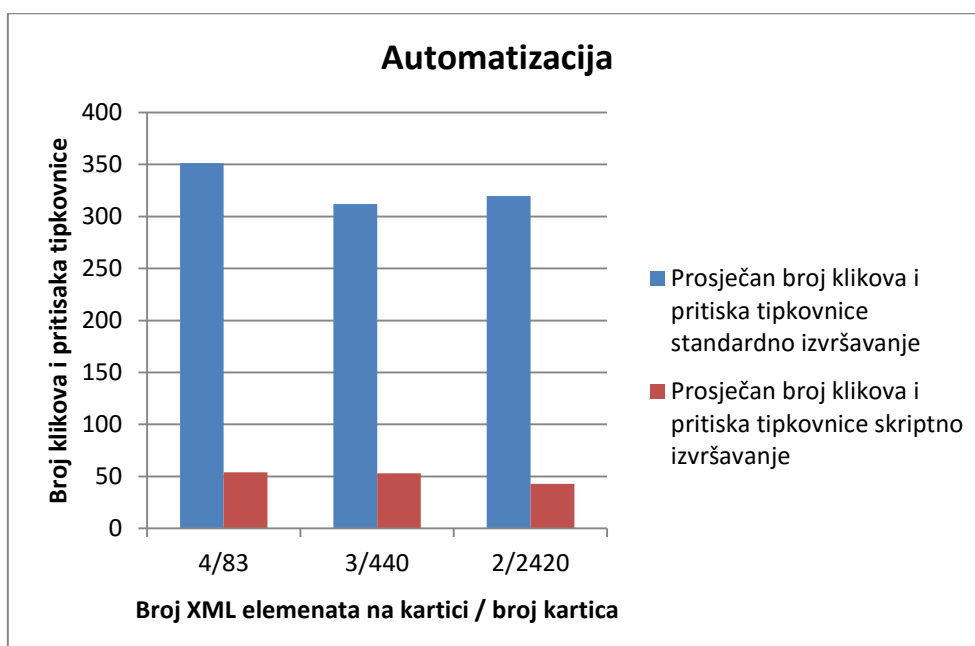
Tablica 24. Usporedbe broja klikova i pritisaka tipkovnice skriptnog i standardnog izvođenja zadatka izrade montaže i personalizacije kartica iz podataka u XML datoteci

Zadatak	Broj XML elemenata na kartici / broj kartica	Prosječan broj klikova i pritisaka tipkovnice standardno izvršavanje	Prosječan broj klikova i pritisaka tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
1.	4/83	351	54	85 %
2.	3/440	312	53	83 %
3.	2/2420	320	43	87 %

Postotak smanjenja broja potrebnih koraka za ostvarenje zadatka izrade pripreme je značajan i možemo zaključiti da je proces dobro automatiziran. Broj koraka neće ovisiti o veličini XML dokumenta, već o montaži koju je potrebno napraviti i broju elemenata koji se postavljaju na kartice. Najveća ušteda u broju koraka ostvarena je kod trećeg zadatka gdje je bilo potrebno napraviti montažu s 25 kartica. Broj koraka kod skriptnog izvođenja zadatka najviše ovisi o broju tekstualnih polja koje je potrebno postaviti na karticu.



Slika 80. Prikaz uštede vremena skriptnim u usporedbi sa standardnim načinom izvođenja zadatka izrade montaže i personalizacije kartica iz podataka u XML datoteci



Slika 81. Prikaz dobivene automatizacije skriptnim u usporedbi sa standardnim načinom izvođenja zadatka izrade montaže i personalizacije kartica iz podataka u XML datoteci

Pri izvođenju zadatka na standardni i skriptni način nije došlo od pogrešaka koje su ostale neispravljene. Pogreške su se ispravljale u trenutku njihovog nastanka. Tablica 25. prikazuje varijacije pri izvođenju zadatka. Iz dobivenih je rezultata vidljivo da testirano standardno izvođenje zadatka ima velike varijacije u broju potrebnih koraka za izvođenje zadatka. Razlog tako velikoj varijaciji u broju potrebnih koraka za izvršavanje zadataka je u kompleksnosti zadatka koji se izrađuje. Kod izrade navedene pripreme ne postoje

ponavljajući zadatci, koraci pripreme se razlikuju i potrebno ih je izvršavati u određenom slijedu. Ukoliko izračunamo pouzdanost prema formuli (7) s ukupnim podacima iz Tablice 25., dobivamo:

$$P(\text{standardno}) = (1 - 0,19)^{328} = 0 \%$$

$$P(\text{skriptno}) = (1 - 0,07)^{50} = 2,7 \%$$

Pouzdanost možemo zaključiti da je vjerojatnost izvršavanja zadatka s minimalnim brojem koraka kod standardnog izvršenja zadatka iznimno mala što proizlazi iz kompleksnosti zadatka. Također je vidljivo da se isti problem pojavljuje i kod skriptnog izvođenja zadatka. Razlog tako niske vjerojatnosti da se zadatak ostvari s minimalnim brojem koraka kod skriptnog izvršavanja zadatka je u tome što sam korisnik pri izvršavanju skripte mora odraditi dosta zadataka. Zadatci koje izvršava korisnik omogućuju da skripta bude varijabilna i da se može primijeniti na većem broju zadataka grafičke pripreme koji uključuju rad s XML podacima.

Tablica 25. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka izrade montaže i personalizacije kartica iz podataka u XML datoteci

Zadatak	1		2		3		Ukupno/ prosjek za sva mjerenja	
Broj XML elemenata na kartici / broj kartica	4/83		3/440		2/2420			
Izvršenje zadatka	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno
Pritisaci miša i tipkovnice								
Minimalan broj (b_{\min})	306	51	260	48	261	41	-	-
Maksimalan broj (b_{\max})	403	58	371	58	386	45	-	-
Raspon varijacije (RV)	97	7	111	10	125	4	-	-
Prosjek (\bar{b})	351	54	312	53	320	43	328	50
Standardna devijacija (σ)	39	3	46	4	50	2	-	-
Koeficijent varijacije (V)	11 %	6 %	15 %	8 %	16 %	4 %	14 %	6 %
Najveće odstupanje od minimalne vrijednosti (V_{\max})	32 %	14 %	43 %	21 %	48 %	10 %	41 %	15 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	15 %	5 %	20 %	10 %	22 %	4 %	19 %	7 %

5.6. Rezultati mjerenja skripte za izradu montaže u programu za prijelom (S6)

Testiranjem skripte za izradu jednostavnih montaža željela se usporediti standardna montaža u programu za prijelom i montaža uz pomoć skripte. Osim izrade jednostavnih montaža, obično se u programu za prijelom (Adobe InDesign) ne montiraju dokumenti, osim u slučajevima jednostavnih montaža, pa su testiranja rađena samo za takve slučajeve. Kako bi se izvršile usporedbe, odabrana su 3 zadatka montaže.

Montaža se na standardan način izrađuje tako da se najprije u program ubacuje PDF dokument kojega je potrebno montirati. Oko ubačenog dokumenta crtaju se rezne oznake te se on montira. Ukoliko je potrebno, pri izradi montaže se ubacuju i dodatne stranice PDF dokumenata koje se trebaju nalaziti na istom arku. Nakon što je gotova montaža prvog arka, ukoliko je potrebno, izrađuju se i ostali arci kopiranjem prvog arka i zamjenom stranica PDF dokumenta.

Skriptni način rada i mogućnosti izrađene skripte opisani su u poglavlju 4.3.6.

Za testiranja su izrađivani sljedeći zadatci:

1. Izrada montaže posjetnica (Slika 82.)

Format: 90 x 50 mm

Broj stranica PDF dokumenta: 3

Montaža: arak 483 x 330 mm, 24 na arku (3 x 8), jednostrano

Napusti: 3 mm

2. Izrada montaže razglednica (Slika 83.)

Format: 150 x 100 mm

Broj stranica PDF dokumenta: 6

Montaža: arak 483 x 330 mm, 9 na arku (3 x 3), obostrano

Napusti: 3 mm

3. Izrada montaže priznanja (Slika 84.)

Format: 210 x 297 mm

Broj stranica PDF dokumenta: 38

Montaža: arak 450 x 320 mm, 2 na arku, jednostrano

Napusti: 3 mm



Slika 82. Izrada montaže posjetnica



Slika 83. Izrada montaže razglednica



Slika 84. Izrada montaže priznanja

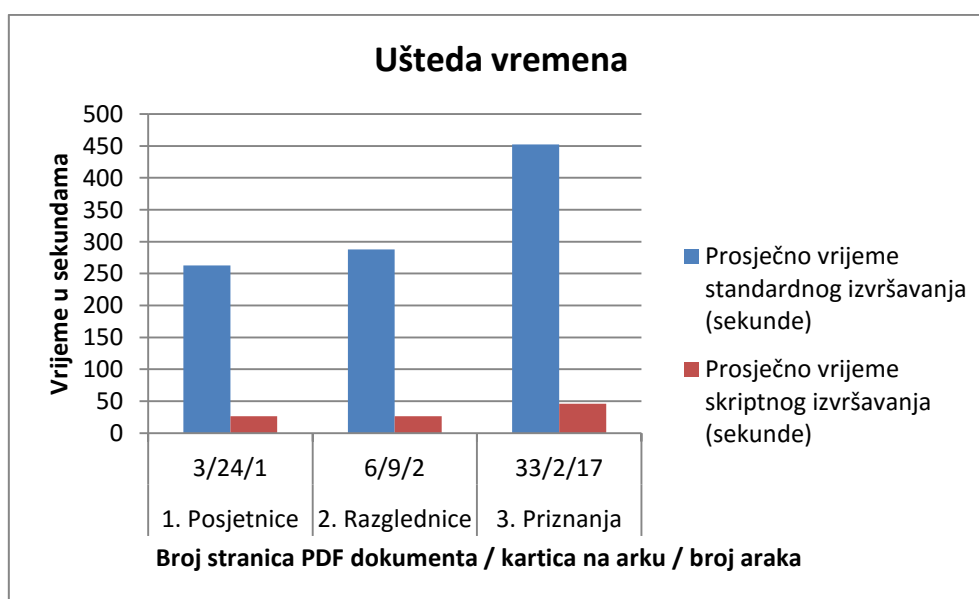
Dobiveni rezultati pojedinačnih mjerenja prikazani su u poglavlju Prilozi u Tablici 48. Tablice 26. i 27. te slike 85. i 86. prikazuju dobivene uštede izračunate iz prosječnih vrijednosti izrađenih mjerenja.

Tablica 26. Usporedbe vremena skriptnog i standardnog izvođenja zadatka izrade montaže u programu za prijelom

Zadatak montaže	Broj stranica PDF dokumenta / kartica na arku / broj araka	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postotci)
1. Posjetnice	3/24/1	263	27	236	90 %
2. Razglednice	6/9/2	288	26	261	91 %
3. Priznanja	33/2/17	452	46	406	90 %

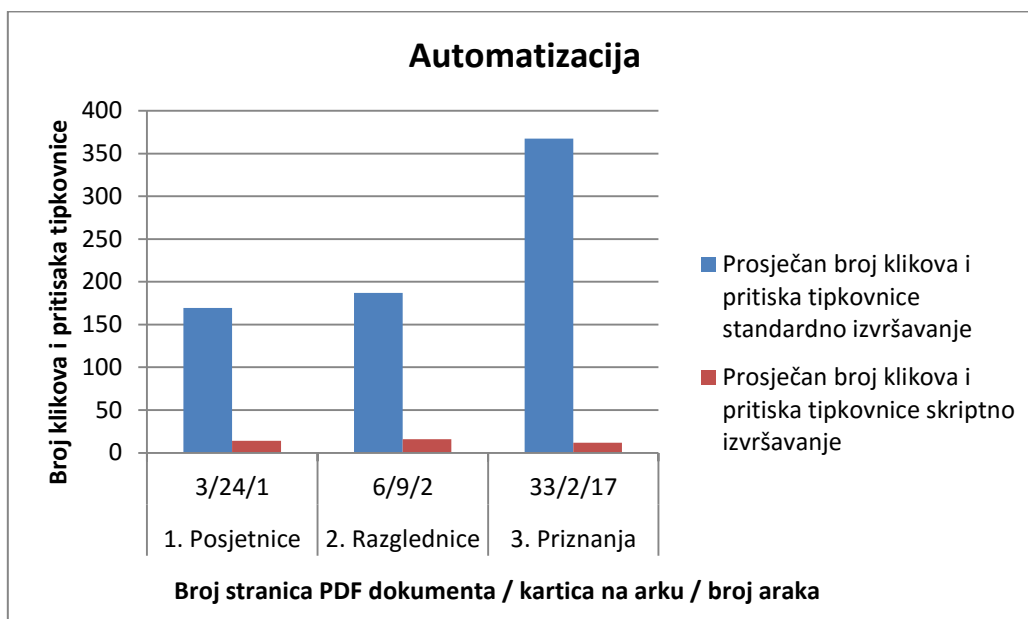
Tablica 27. Usporedbe broja klikova i pritisaka tipkovnice skriptnog i standardnog izvođenja zadatka izrade montaže u programu za prijelom

Zadatak montaže	Broj stranica PDF dokumenta / kartica na arku / broj araka	Prosječan broj klikova i pritisaka tipkovnice standardno izvršavanje	Prosječan broj klikova i pritisaka tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
1. Posjetnice	3/24/1	170	14	92 %
2. Razglednice	6/9/2	187	16	92 %
3. Priznanja	33/2/17	367	12	97 %



Slika 85. Prikaz uštede vremena skriptnim u usporedbi sa standardnim načinom izvođenja zadatka izrade montaže u programu za prijelom

Dobiveni rezultati prikazuju veliku uštedu u vremenu rada i u postignutoj automatizaciji. Vrijeme izrade montaže smanjeno je u prosjeku za 90 % i postignut je vrlo visok nivo automatizacije. Veći nivo automatizacije postignut je kod poslova montaže koji zahtijevaju rad s više stranica koje se montiraju. Tako će složeniji poslovi montaže donositi veću automatizaciju.



Slika 86. Prikaz dobivene automatizacije skriptnim u usporedbi sa standardnim načinom izvođenja zadatka izrade montaže u programu za prijelom

Pri izvođenju zadatka na standardan i skriptni način nije došlo od pogrešaka koje su ostale neispravljene. Pogreške su se ispravljale u trenutku njihovog nastanka. Tablica 28. prikazuje varijacije pri izvođenju zadatka. Ukoliko izračunamo pouzdanost prema formuli (7) s ukupnim podacima iz Tablice 28., dobivamo:

$$P(\text{standardno}) = (1 - 0,23)^{241} = 0 \%$$

$$P(\text{skriptno}) = (1 - 0,10)^{14} = 22,9 \%$$

Pouzdanost možemo zaključiti da je vjerojatnost izvršavanja zadatka s minimalnim brojem koraka kod standardnog izvršenja iznimno mala što proizlazi iz kompleksnosti zadatka. Kod skriptnog načina izvođenja zadatka imamo puno manje varijacije u broju potrebnih koraka i možemo zaključiti da je veća vjerojatnost da će se zadatak optimalno izvršiti upotrebom skriptnih tehnologija, jer je potreban uvelike manji broj koraka, koji se ostvaruje s mnogo manje varijacija.

Tablica 28. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka izrade montaže u programu za prijelom

Zadatak	1. Posjetnice		2. Razglednice		3. Priznanja		Ukupno/ prosjek za sva mjerenja	
Broj stranica / kartica na arku / broj araka	3/24/1		6/9/2		33/2/17			
Izvršenje zadatka	standardno	skriptno	standardno	skriptno	standardno	skriptno	standardno	skriptno
Pritisci miša i tipkovnice								
Minimalan broj (b_{\min})	141	12	140	15	319	11	-	-
Maksimalan broj (b_{\max})	241	17	257	17	436	15	-	-
Raspon varijacije (RV)	100	5	117	2	117	4	-	-
Prosjek (\bar{b})	170	14	187	16	367	12	241	14
Standardna devijacija (σ)	43	2	45	1	48	2	-	-
Koeficijent varijacije (V)	25 %	14 %	24 %	7 %	13 %	14 %	21 %	12 %
Najveće odstupanje od minimalne vrijednosti (V_{\max})	71 %	42 %	84 %	13 %	37 %	36 %	64 %	30 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	20 %	17 %	34 %	5 %	15 %	9 %	23 %	10 %

5.7. Rezultati mjerenja skripte za automatizaciju izrade ePub dokumenta koji sadrži SVG datoteke (S7)

Adobe InDesign pri izradi grafičke pripreme predviđa potrebu izrade ePub dokumenta, no njegove novije verzije ne podržavaju upotrebu SVG datoteka. Ukoliko se iz prelomljenih knjiga želi dobiti ePub dokument koji sadrži SVG datoteke, one se moraju dodati naknadno u već izrađen ePub dokument.

Kako je postupak naknadne izrade i povezivanja SVG datoteka u ePub dokumentu dugotrajan, željelo se skratiti vrijeme izrade i automatizirati rad.

Automatizacija procesa i izrada skripte rađena je prema standardnom načinu izrade. Standardni način izrade ePub dokumenta koji sadrži SVG datoteke rađen je kako slijedi:

- U dokumentu pripremljenom za izvoženje u ePub potrebno je izabrati svaku EPS ili AI datoteku koja će se zamijeniti SVG datotekom. Pri odabiru datoteka na stranicama dokumenta se selektira okvir koji sadrži datoteku te se postavlja *Object Export Options/ Alt Text* na naziv SVG datoteke koja će se kasnije povezati. Postavljanje *Alt Text* opcije olakšava kasnije lakše pronalaženje poveznica u XHTML datoteci ePub dokumenta i olakšava izmjenu izvezene slike sa SVG datotekom. Pri izvoženju dokumenta u ePub javlja se problem, tj. Adobe InDesign sve poveznice koje su nepodržane u ePub dokumentu pretvara u bitmape kojima mijenja naziv. Tako je kasnije teško pronaći novoimenovane datoteke i zamijeniti ih SVG datotekama. Ukoliko se u *Alt Text* zapiše naziv SVG datoteke s kojom je potrebno napraviti supstituciju, olakšava se kasniji zadatak izmjene.
- Izrada SVG datoteka iz AI i EPS datoteka. Svaka datoteka se zasebno otvara u programu Adobe Illustrator te se prije spremanja uređuje sljedećim koracima:
 - *Flatten Transparency* (kako ne bi nastali problemi s prikazom SVG datoteka koje sadrže transparencije. SVG datoteke s transparencijama često su nepodržane u ePub čitaćima)
 - *Proporcionalno skaliranje crteža na 133.333 %* (Illustrator izvozi SVG datoteke pod pretpostavkom rezolucije uređaja od 72 dpi. Budući da većina ePub čitača podrazumijeva rezoluciju od 96 dpi, potrebno je skalirati SVG dokument kako bi se isti dobro prikazivao u ePub čitaćima)
 - *Prilagođavanje Artboarda na veličinu crteža*
 - *Spremanje SVG datoteke prema nazivu zadanom u Alt Text*

- Otpakiravanje ePub dokumenta i zamjena bitmapa SVG datotekama. Mijenjanje poveznica u .xhtml datoteci ePub dokumenta.

Prema opisanom načinu izvršavana je standardna izrada dok je skriptni način opisan u poglavlju 4.3.7. Dobiveni rezultati svih mjerenja prikazani su u poglavlju Prilozi u Tablici 49. Pri mjerenju vremena i broja pritisaka tipki tipkovnice i miša nije uključena priprema dokumenta za izvoženje u ePub dokument.

Prosječni rezultati za izrađena mjerenja prikazani su u Tablicama 29. i 30. te na Slikama 87. i 88. Iz dobivenih rezultata vidljivo je da se upotrebom skriptnih tehnologija u ostvarivanju zadatka ostvaruju visoke uštede u vremenu izrade i u automatizaciji procesa. Također je vidljivo da se povećanjem broja poveznica povećavaju dobivene uštede. Skriptno vrijeme, broj klikova i pritisaka tipkovnice nisu se uvelike razlikovali ovisno o broju poveznica koje je bilo potrebno zamijeniti dok je kod standardnog načina izrade došlo do većih odstupanja, pa tako i većih ušteda, pri povećanju broja poveznica.

Izrađena skripta omogućuje automatiziranu izradu procesa kojega je teško ostvarivati zbog nedostatka podrške u programu. Zbog nedostatka podrške za rad sa SVG datotekama otežan je proces rada sa SVG dokumentima te je upotrebom skriptnih tehnologija moguće dobiti visoku automatizaciju i uštedu vremena. Skripta prikazuje kako je moguće povezivati programe grafičke pripreme skriptnim tehnologijama i olakšati rad u više programa preko jedne skripte.

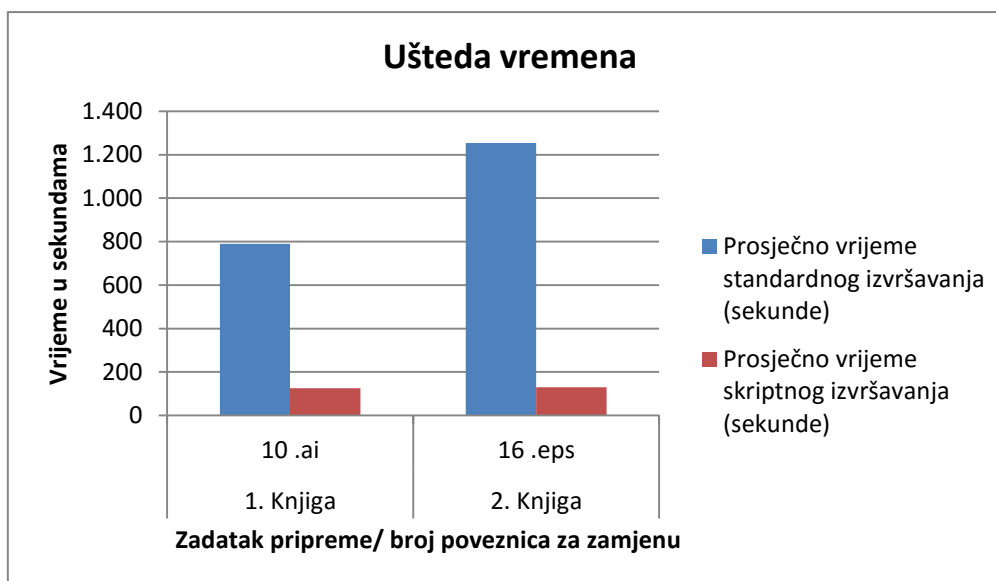
Dobiveni rezultati potvrđuju da je zamjenom nestandardnih radnih procesa sa skriptnim tehnologijama moguće automatizirati procese grafičke pripreme i rješavati zadatke koji su inače teški za ostvarivanje.

Tablica 29. Usporedbe vremena skriptnog i standardnog izvođenja zadatka izrade ePub dokumenta koji sadrži SVG datoteke

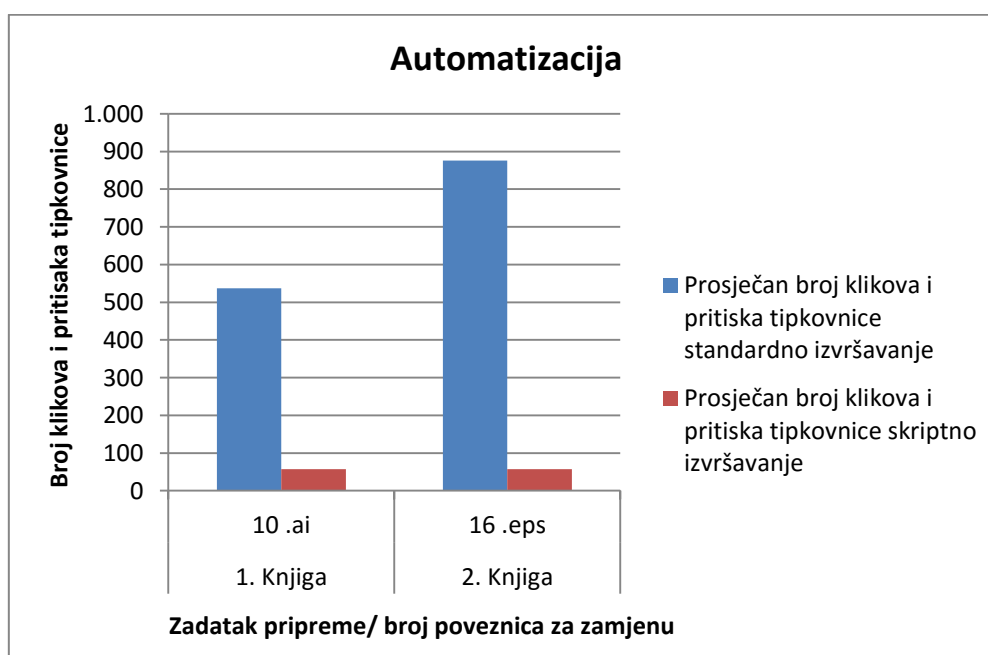
Zadatak	Broj poveznica za zamjenu	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postotci)
1. Knjiga	10.ai	789	125	664	84 %
2. Knjiga	16.eps	1.254	130	1123	90 %

Tablica 30. Usporedbe broja klikova i pritisaka tipkovnice skriptnog i standardnog izvođenja zadatka izrade ePub dokumenta koji sadrži SVG datoteke

Zadatak	Broj poveznica za zamjenu	Prosječan broj klikova i pritisaka tipkovnice standardno izvršavanje	Prosječan broj klikova i pritisaka tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
1. Knjiga	10.ai	537	58	89 %
2. Knjiga	16.eps	876	57	93 %



Slika 87. Prikaz uštede vremena skriptnim u usporedbi sa standardnim načinom izvođenja zadatka izrade ePub dokumenta koji sadrži SVG datoteke



Slika 88. Prikaz dobivene automatizacije skriptnim u usporedbi sa standardnim načinom izvođenja zadatka izrade ePub dokumenta koji sadrži SVG datoteke

Tablica 30. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka izrade ePub dokumenta koji sadrži SVG datoteke

Zadatak	1. Knjiga		2. Knjiga		Ukupno/ prosjeak za sva mjerenja	
Broj poveznica za zamjenu	10.ai		16.eps			
Izvršenje zadatka	standardno	skriptno	standardno	skriptno	standardno	skriptno
Pritisci miša i tipkovnice						
Minimalan broj (b_{\min})	498	52	837	52	-	-
Maksimalan broj (b_{\max})	598	62	942	65	-	-
Raspon varijacije (RV)	100	10	105	13	-	-
Prosjeak (\bar{b})	537	58	876	57	706	58
Standardna devijacija (σ)	39	4	43	5	-	-
Koeficijent varijacije (V)	7 %	7 %	5 %	9 %	6 %	8 %
Najveće odstupanje od minimalne vrijednosti (V_{\max})	20 %	19 %	13 %	25 %	16 %	22 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	8 %	11 %	5 %	10 %	6 %	11 %

Pri izvođenju zadatka na standardni i skriptni način nije došlo od pogrešaka koje su ostale neispravljene. Pogreške su se ispravljale u trenutku njihovog nastanka. Tablica 31. prikazuje varijacije pri izvođenju zadatka. Ukoliko izračunamo pouzdanost prema formuli (7) s ukupnim podacima iz Tablice 30., dobivamo:

$$P(\text{standardno}) = (1 - 0,06)^{706} = 0 \%$$

$$P(\text{skriptno}) = (1 - 0,11)^{58} = 0,11 \%$$

Pouzdanost možemo zaključiti da je vjerojatnost izvršavanja zadatka s minimalnim brojem koraka kod standardnog izvršenja zadatka iznimno mala što proizlazi iz kompleksnosti zadatka i pogrešaka koje često nastaju pri njegovu izvođenju. Kod skriptnog načina izvođenja također postoje varijacije jer dijelove zadatka odrađuje korisnik, ali će pouzdanost izvršavanja zadatka s minimalnim brojem koraka biti veća zbog uvelike smanjenog broja koraka. Ukoliko usporedimo potreban broj koraka za skriptno i standardno izvršavanje zadatka, vidljivo je da je postignuta velika automatizacija pa će iz navedenog proizlaziti da se zadatak može skriptno optimalno izvršavati.

5.8. Rezultati mjerenja skripte za automatsku obradu fotografija (S8)

Tipično podešavanje slika koje se pripremaju za tisak u programu Adobe Photoshop opisano je u Poglavlju 4.3.8.

Standardno izvođenje zadatka ostvaruje se preko *Batch* opcije programa Adobe Photoshop. Navedenom opcijom izvršavaju se gotovo svi zadatci uređivanja slike osim povećavanja na željenu rezoluciju. Povećavanje slike ovisi o njezinoj veličini u programu Adobe InDesign (ovisno o *effective ppi*). Povećavanje slike na željenu rezoluciju potrebno je obaviti prije obrade pa je *Batch* opcijom moguće urediti samo one slike koje su zadovoljavajuće rezolucije. *Batch* opcija povezuje akciju koja je spremljena pod paletom *Actions* u programu Adobe Photoshop i primjenjuje ju na otvorene slike ili selektiranu mapu. Kako slike ponovnim spremanjem u JPEG format pri kompresiji gube podatke (*eng. Generation Loss*) [63], spremaju se kao PSD datoteke. Spremljene PSD datoteke potrebno je ubaciti u Adobe InDesign upotrebom *Relink* opcije.

Standardno izvršavanje zadatka uključuje sljedeće korake:

1. Odabir slika u *Links* paleti Adobe InDesigna koje su preniske rezolucije i otvaranje u Adobe Photoshop programu preko *opcije Edit With → Adobe Photoshop CS6*.
2. Povećavanje slika na zadovoljavajući *epi* preko spremljene akcije koja se provodi više puta ovisno o veličini slike (povećanje slike u koracima od 110 %). Nakon zadovoljavajućeg povećavanja pokreće se akcija obrade slike. Naposljetku se slika sprema kao PSD datoteka.
3. Povezivanje PSD datoteka u Adobe InDesign programu preko *Relink* opcije.
4. Otvaranje svih slika koje su zadovoljavajuće rezolucije, ali u RGB prostoru boja, kako bi se provela *Batch* akcija obrade svih slika i spremanje u PSD formatu.
5. Mijenjanje poveznica u dokumentu s novonastalim PSD datotekama.

Skripta obrađuje slike na isti način i u konačnici se dobiva gotovo isti dokument. Izvršavanje skripte opisano je u poglavlju 4.3.8. Skripta koja uređuje slike u Adobe Photoshopu izrađena je pomoću dodatka *ScriptListener* koji olakšava skriptiranje [64]. Navedeni dodatak zapisuje svaki korak izrađen u programu Adobe Photoshop u tekstualnu datoteku u obliku skriptnog koda (izrađuju se 2 datoteke, jedna sadrži JavaScript, a druga VBScript kod) i na taj način olakšava konačnu izradu skripte. Naime, nije potrebno pisati kod, već ga je dovoljno kopirati iz nastalih datoteka.

Testiranje automatizacije i vremena izvršavanja rađeno je na dva dokumenta. Dokumenti se razlikuju u broju poveznica koje je potrebno obraditi. Tablica 31. opisuje obrađivane dokumente.

Tablica 31. Opis dokumenata korištenih za testiranje

<p>1. Brošura Broj poveznica: 17 Broj JPEG slika: 13 RGB Slike s <i>effective ppi</i> manjim od 300: 6 JPEG Poveznice koje se ne obrađuju: 2 PDF, 2 PSD</p>
<p>2. Časopis Broj poveznica: 108 Broj JPEG slika: 64 RGB + 1 c/b + 10 CMYK Slike s <i>effective ppi</i> manjim od 300: 31 JPEG Poveznice koje se ne obrađuju: 2 AI, 20 PDF, 11 PSD</p>

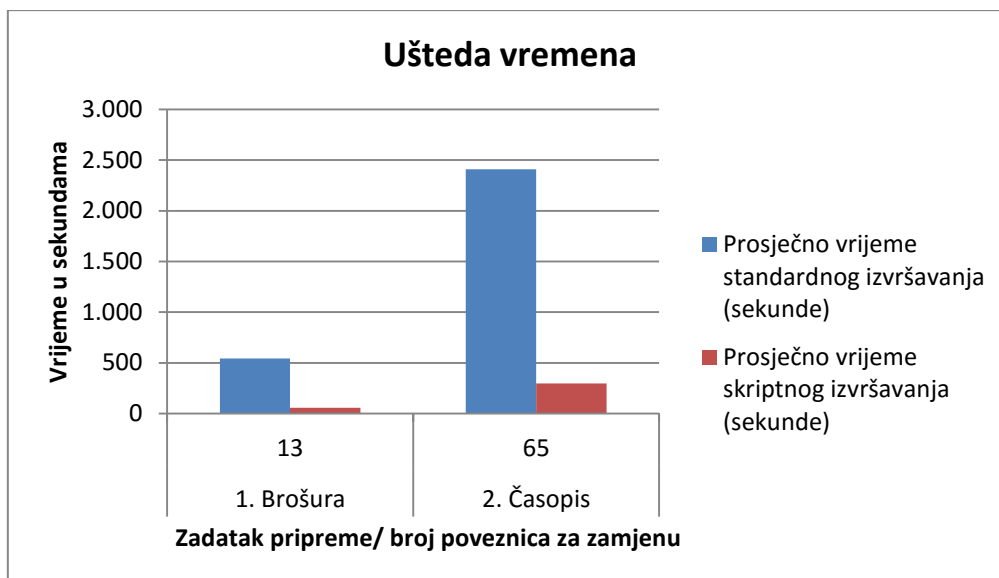
Dobiveni rezultati pojedinačnih mjerenja prikazani su u Tablici 50. Iz rezultata su izvedeni prosjeci za sva mjerenja i prikazani kao uštede vremena i postignuta automatizacija u Tablicama 32. i 33. te na Slikama 89. i 90. Rezultati pokazuju visok stupanj automatizacije procesa i veliku uštedu vremena. Izvedeno testiranje potvrđuje kako je svakodnevnne zadatke moguće uvelike optimizirati skriptnim tehnologijama.

Tablica 32. Usporedbe vremena skriptnog i standardnog izvođenja zadatka obrade fotografija

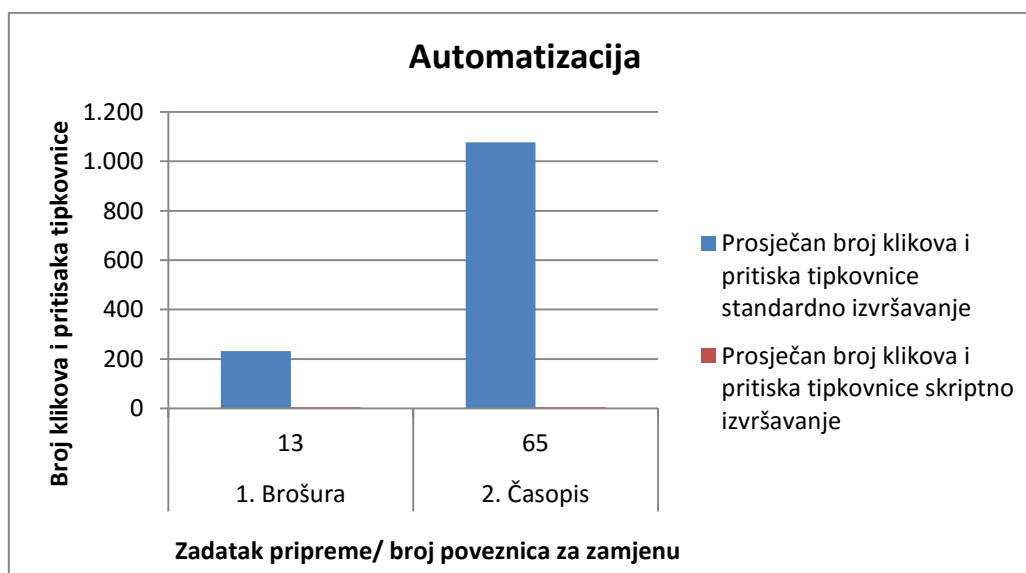
Zadatak	Broj poveznica za zamjenu	Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Ušteda vremena (sekunde)	Ušteda u vremena (postotci)
1. Brošura	13	544	56	488	90 %
2. Časopis	65	2.409	295	2114	88 %

Tablica 33. Usporedbe broja klikova i pritisaka tipkovnice skriptnog i standardnog izvođenja zadatka obrade fotografija

Zadatak	Broj poveznica za zamjenu	Prosječan broj klikova i pritisaka tipkovnice standardno izvršavanje	Prosječan broj klikova i pritisaka tipkovnice skriptno izvršavanje	Postotak smanjenja broja potrebnih koraka
1. Brošura	13	232	5	98 %
2. Časopis	65	1.076	5	100 %



Slika 89. Prikaz uštede vremena skriptnim u usporedbi sa standardnim načinom izvođenja zadatka obrade fotografija



Slika 90. Prikaz dobivene automatizacije skriptnim u usporedbi sa standardnim načinom izvođenja zadatka obrade fotografija

Nakon provjere svih izrađenih priprema, u dokumentima nisu pronađene pogreške koje su ostale neispravljene. Pogreške su se u standardnom načinu izvođenja ispravljale kako su nastajale, što je vidljivo u varijaciji koraka pri standardnom izvođenju zadatka u Tablici 34. Kod skriptnog izvođenja zadatka nema varijacija u broju koraka jer korisnik treba samo pokrenuti skriptu i odabrati željeni profil. Iz navedenog se može zaključiti da će se skriptno izvršavanje zadatka gotovo uvijek optimalno izvršavati.

Tablica 34. Varijacije u broju pritisaka miša i tipkovnice standardnog i skriptnog izvođenja zadatka obrade fotografija

Zadatak	1. Brošura		2. Časopis		Ukupno/ prosjek za sva mjerenja	
Broj poveznica za zamjenu	13		65			
Izvršenje zadatka	standardno	skriptno	standardno	skriptno	standardno	skriptno
Pritisci miša i tipkovnice						
Minimalan broj (b_{\min})	187	5	1032	5	-	-
Maksimalan broj (b_{\max})	287	5	1117	5	-	-
Raspon varijacije (RV)	100	0	85	0	-	-
Prosjek (\bar{b})	232	5	1076	5	654	5
Standardna devijacija (σ)	36	0	32	0	-	-
Koeficijent varijacije (V)	16 %	0 %	3 %	0 %	9 %	0 %
Najveće odstupanje od minimalne vrijednosti (V_{\max})	53 %	0 %	8 %	0 %	31 %	0 %
Prosječno odstupanje od minimalne vrijednosti (\bar{V})	24 %	0 %	4 %	0 %	14 %	0 %

Iz podataka u Tablici 34. možemo izračunati pouzdanosti optimalnog izvršavanja zadatka koje iznose:

$$P(\text{standardno}) = (1 - 0,14)^{654} = 0 \%$$

$$P(\text{skriptno}) = (1 - 0)^5 = 100 \%$$

Testirani zadatak obrade fotografija i pripreme za tisak može se optimalno i u minimalnom broju koraka izvršavati samo skriptno. Kod standardnog izvršavanja zadatka često se događaju pogreške pri radu koje uvjetuju veće razlike u broju potrebnih koraka za izvršavanje zadatka. Pri izvođenju testova nije došlo do pogrešaka koje su ostale neispravljene, ali su one moguće zbog velikog broja koraka koje je potrebno izraditi.

5.9. Rezultati skripte za izradu barkoda iz varijabilnih podataka (S9)

Skripte opisane u Poglavljima 5.3.9 i 5.3.10 razlikuju se od ostalih skripti jer za njih ne postoji optimalan standardan način izvršavanja zadatka. Izrađene skripte izvršavaju zadatke koji su zahtjevni, teški i neisplativi za ostvarivanje u standardnim programima grafičke pripreme. Navedene skripte omogućuju nove modele rada i nove mogućnosti grafičke pripreme u postojećim programima.

Testiranjem skriptnog načina izvođenja zadataka skripte za izradu barkoda iz varijabilnih podataka željelo se potvrditi zadovoljava li izrađeni model potrebe izrade digitalne grafičke pripreme s varijabilnim podacima te su odabrani ispitanici ocjenjivali novu skriptno dodanu mogućnost.

Kako bi se testiralo izvršavanje zadatka skriptnim tehnologijama, ukupno su izrađena 3 testa s 5 ispitanika sličnih predznanja u području grafičke pripreme. Prije testiranja pet je odabranih ispitanika odgovaralo na anketni upitnik. Upitnik je naveden u nastavku zajedno s dobivenim odgovorima:

1. Biste li u programu Adobe InDesign izrađivali varijabilnu grafičku pripremu koja uključuje izradu montaže, numeraciju i izradu Code 128 barkoda na veći broj kartica?
 - a. Ne - 100 %
2. Što smatrate glavnim problemom izrade navedene pripreme u programu Adobe InDesign?

Odgovori:

Ispitanik 1: Dugotrajna priprema koja je neisplativa.

Ispitanik 2: Neekonomičan proces kojega je lakše ostvariti programima za VDP, problem naplate utrošenog vremena za izradu.

Ispitanik 3: Vremenski zahtjevna priprema prilikom koje postoji velika mogućnost pogreške. Nemogućnost provjere ispravnosti postavljenih kodova.

Ispitanik 4: Kako InDesign ne omogućava izradu barkoda, iznimno je teško i zahtjevno ubacivati veliku količinu kodova na tražena mjesta. Postoji i problem izrade kodova jer bi trebalo koristiti dodatne alate.

Ispitanik 5: Priprema bi trajala predugo.

Na Slici 91. prikazano je prosječno vrijeme potrebno za izradu 3 pripreme od strane svakog ispitanika gdje je varijacija bila u broju kartica koje je bilo potrebno izraditi. Za mjerenja su korištene sljedeće postavke zadatka:

Veličina tiskovnog arka: 483 x 330 mm;

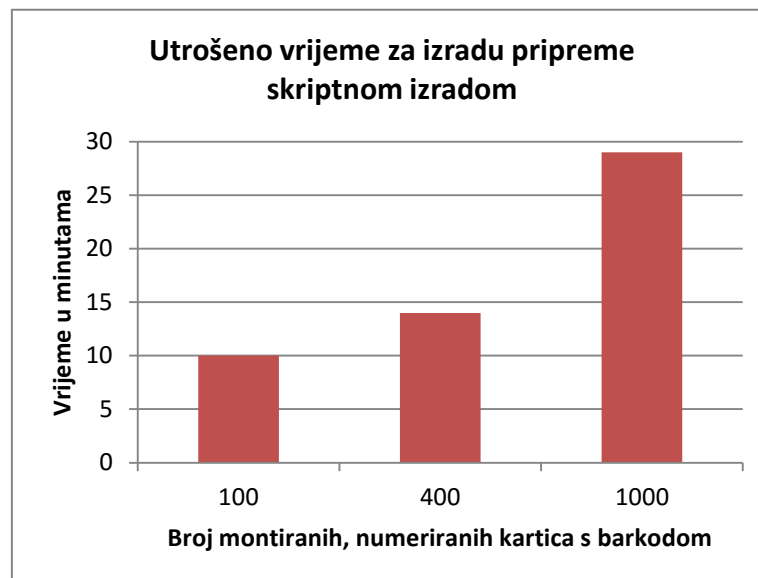
Veličina PDF dokumenta: 205 x 60 mm, 17.7 MB;

Montaža: glava gore; 8 kartica na arku; numeracija i barkod se povećavaju kroz arke kako bi se nakon rezanja dokumenta dobili sabrani blokovi;

Napusti: 3 mm;

Broj vodećih nula: 6;

Broj kartica za montažu, numeriranih polja i barkodova (Code 128 B): **100, 400 i 1000.**



Slika 91. Utrošena vremena za izradu grafičke pripreme koja uključuje montažu, numeraciju i izradu barkoda upotrebom izrađenih skripti na 100, 400 i 1000 kartica

Nakon testiranja skripti postavljen je novi upitnik. Pitanja i dobiveni odgovori su kako slijedi:

1. Ocjenom od 1 do 5 ocijenite dodanu mogućnost u programu:
 - a. Ocjena 5 – 100 %
2. Smatrate li da su skriptnim tehnologijama uklonjeni ranije navedeni nedostaci u izradi tražene pripreme?
 - a. Da – 100 %

Mjerenja su rađena na dokumentu prikazanom u poglavlju 4.3.9. Iz dobivenih rezultata mjerenja vremena, kako je prikazano na Slici 91., može se zaključiti da se mogu ostvariti značajne uštede u vremenu izrade upotrebom skriptiranja u zadacima koji nisu predviđeni za izradu u standardnim programima za grafičku pripremu, tj. da skriptiranje omogućuje zadatke koji bi inače bili neisplativi.

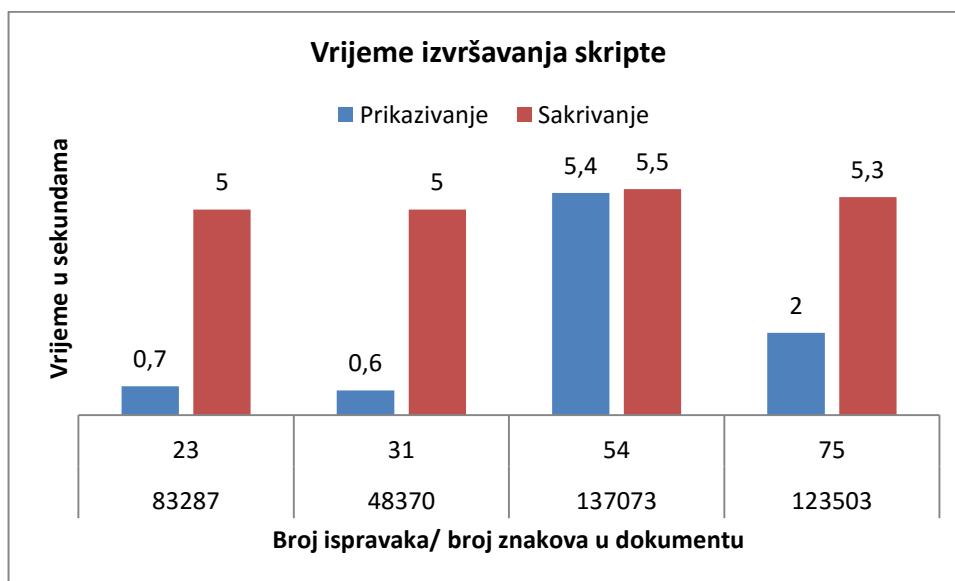
Svi su ispitanici procijenili da u Adobe InDesign programu manualna izvedba zadatka nije moguća zbog velikog broja koraka koji bi se trebali odraditi te problema nečitljivosti barkoda što uzrokuje nemogućnost vizualne procjene ispravnosti pripreme. Pogreške koje bi nastale u pripremi teško bi se provjeravale osim da se čitačem barkoda provjeri svaki izrađeni kod. Pouzdanost same skripte nije uzeta u obzir jer ukoliko je skripta dovoljno testirana i optimizirana, vjerojatnost pogreške je izuzetno mala. Za testiranu skriptu ni u jednom testu nije došlo do pogreške u skripti pa se utjecaj skripte na pouzdanost izvršavanja zadatka zanemaruje. Pogreške u izvršavanju skripte češće bi se mogle dogoditi ukoliko korisnik odredi pogrešne ulazne parametre, dakle veća je vjerojatnost da će nastati pogreška zbog ljudskog faktora.

5.10. Rezultati skripte za obilježavanje ispravaka u tekstu nakon lekture (S10)

Skripta za obilježavanje ispravaka u tekstu nakon lekture izrađena je kako bi se dodala nova mogućnost u postojeći program preko skriptiranja. Odluka za izradu navedene skripte donesena je praćenjem grafičkog procesa. Skripta je implementirana u proizvodni proces i ocijenjena primjenjivom jer omogućuje jednostavno digitalno odobravanje izrađenih ispravaka u dokumentu za objavu. Skripta osim uštede vremena odobravanja štedi i resurse jer ispravke nije potrebno slati naručiocu na ispisanom mediju. Samo vrijeme izvršavanja skripte je kratko što se može vidjeti na Slici 92.

Testiranja su rađena na pričama različitih dužina i različitog broja ispravaka. Vrijeme je mjereno uz pomoć \$.hiresTimer opcije programa InDesign.

Izrađena skripta dokazuje kako je skriptiranjem moguće implementirati nove mogućnosti u program koje ranije nisu postojale. Svrha uvođenja novih mogućnosti je olakšavanje rada i optimizacija proizvodnih procesa.



Slika 92. Utrošena vremena za prikazivanje i sakrivanje ispravaka zabilježenih preko *Track Changes* opcija

6. DISKUSIJA

6.1. Optimizacija vremena izvršavanja zadataka grafičke pripreme

U Tablici 35. i na Slici 93. prikazani su rezultati izrađenih mjerenja te ukupne uštede vremena pri skriptnom izvršavanju u usporedbi sa standardnom izradom zadatka. Iz dobivenih rezultata vidljivo je da se upotrebom skripti pri rješavanju zadataka digitalne grafičke pripreme može postići velika ušteta u vremenu. Ušteta vremena prema standardnoj izradi iskazana u postotcima kreće se od 62 % do 97 %, a u prosjeku iznosi 87 %.

Iz izrađenih mjerenja može se zaključiti da ušteta vremena izrade digitalne grafičke pripreme skriptnim načinom ovisi o opsegu zadatka koji se izvršava, zadatku kojega skripta mijenja, korisniku skripte koji može biti uključen u izvršavanje procesa te načinu kako je skripta napisana.

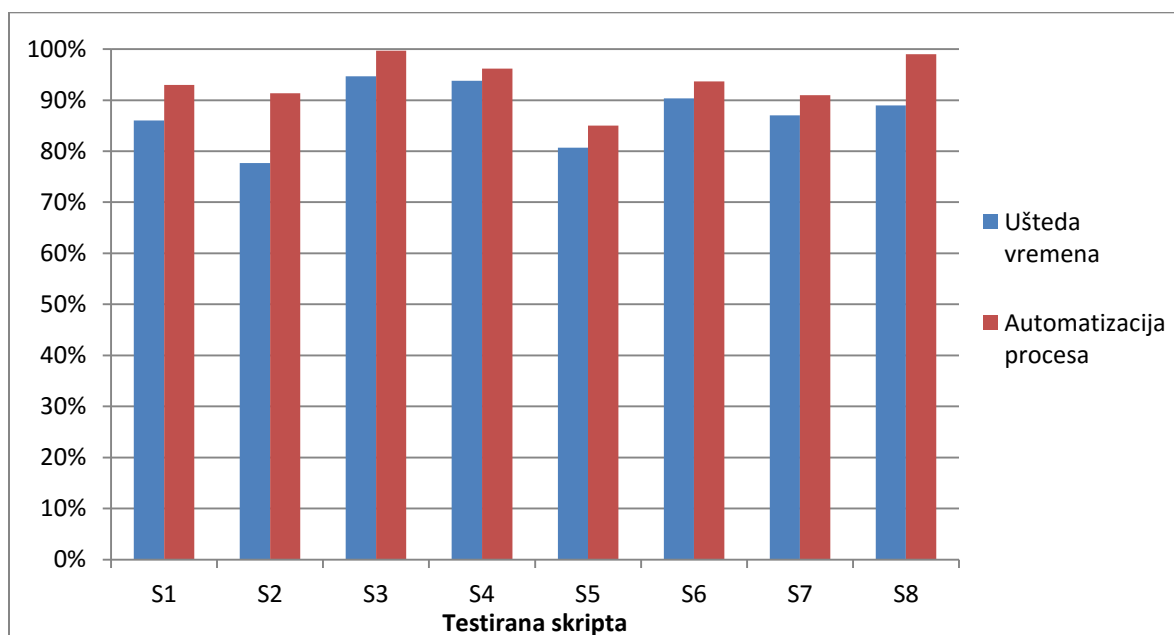
Postotak uštede vremena pri testiranju pojedinačne skripte ovisi o veličini dokumenta koji se uređuje. Gotovo je u svim mjerenjima skriptnog izvršavanja, u usporedbi sa standardnom izradom, povećanjem opsega zadatka došlo do smanjenja postotka uštede. To je osobito vidljivo kod zadataka koji imaju kod standardne izrade mogućnost optimizacije i automatizacije procesa preko opcija izbornika programa (S1, S2, S4, S5). U navedenim je slučajevima udio vremena rada korisnika manji kako se povećava veličina dokumenta pa se smanjuje i postotak vremenske uštede.

Kod skripte S3 također dolazi do smanjenja postotka vremenske uštede povećanjem opsega zadatka, ali ne iz istih razloga. Kod standardnog izvođenja zadatka skripte S3 ne postoji automatizacija preko izbornika programa. Jedan od razloga smanjenja postotka vremenske uštede pri povećanju opsega dokumenta pri izvršavanju zadatka skripte S3 je način kako se zadatak standardno izvršava i vrijeme za izvršenje koje je potrebno korisniku da izvrši zadatak. U slučaju kad imamo kratke tablice koje se uređuju, korisnik mora postaviti mnogo različitih postavki, a malo ponavljajućih. Ponavljajući zadatci koji nisu složeni, brže se standardno izvode od kompleksnih promjenjivih zadataka. Tako je kod najkraće testirane tablice od 145 redaka za prijelom bilo potrebno oko 5 sekundi po retku, a kod najduže tablice od 1103 retka 2 sekunde po retku. Povećanjem veličine tablice udio je vremena oblikovanja cijele tablice u ukupnom vremenu manji, a raste udio vremena za oblikovanje pojedinačnih redaka tablice. Time se povećava udio ponavljajućih zadataka pa se povećava brzina rada i skraćuje standardno vrijeme izvršenja zadatka.

U dobivenim sumarnim rezultatima vidljiv je utjecaj zadatka kojega skripta mijenja na dobivene rezultate pa je gotovo svaka skripta imala drugačiji postotak optimizacije vremena. Zadatci za koje pri standardnom izvođenju postoji alternativa u rješavanju djelomičnom automatizacijom preko izbornika u programu, dat će nešto manju uštedu vremena dok se zadatci za koje nije predviđena automatizacija u programu, mogu bolje vremenski optimizirati upotrebom skriptnih tehnologija.

Tablica 35. Dobivene ukupne uštede vremena skriptnog izvršavanja zadatka u usporedbi sa standardnom izradom po testiranim skriptama

	Testirani zadatci	1.	2.	3.	4.	5.	Prosjek
S1	Brisanje neželjenih bjelina u tekstu	94 %	93 %	84 %	73 %		86 %
S2	Zadržavanje uvezenih stilova	93 %	78 %	62 %			78 %
S3	Uređivanje tablice u periodičkoj publikaciji	97 %	95 %	92 %			95 %
S4	Numeracija i montaža kartica	97 %	95 %	94 %	92 %	91 %	94 %
S5	Izrada montaže i personalizacije iz XML datoteke	84 %	85 %	73 %			81 %
S6	Izrada jednostavnih montaža	90 %	91 %	90 %			90 %
S7	ePub dokument sa SVG datotekama	84 %	90 %				87 %
S8	Automatska obrada fotografija	90 %	88 %				89 %
Prosjek:							87 %



Slika 93. Dobivena ušteda vremena i automatizacija procesa u postotcima po testiranim skriptama

Tablica 36. Brzina standardnog i skriptnog izvođenja u ovisnosti o povećanju opsega zadatka

Zadatak		Prosječno vrijeme standardnog izvršavanja (sekunde)	Prosječno vrijeme skriptnog izvršavanja (sekunde)	Brzina u sekundama po	
S1				broju znakova u dokumentu	
Knjiga 1	39.398	80	5	0,00203	0,00013
Knjiga 2	60.914	81	6	0,00133	0,00010
Knjiga 3	399.258	89	14	0,00022	0,00004
Knjiga 4	789.126	102	28	0,00013	0,00004
S2				broju znakova u dokumentu	
Knjiga 1	55.216	71	5	0,00129	0,00009
Knjiga 2	398.360	78	17	0,00020	0,00004
Knjiga 3	787.466	95	36	0,00012	0,00005
S3				broju redaka tablice	
Tablica 1	145/9	687	21	4,74	0,14
Tablica 2	573/9	1194	59	2,08	0,10
Tablica 3	1103/15	2210	186	2,00	0,17
S4				broju tekstualnih okvira na kartici	
1		980	28	980,00	28,00
2		1113	55	556,50	27,50
3		1327	84	442,33	28,00
4		1482	116	370,50	29,00
5		1637	154	327,40	30,80
S5				broju XML elemenata	
1.	4/83	543	85	1,64	0,26
2.	3/440	649	98	0,49	0,07
3.	2/2420	967	265	0,20	0,05
S6				broju montiranih kartica i araka	
1. Posjetnice	3/24/1	263	27	3,65	0,38
2. Razglednice	6/9/2	288	26	2,67	0,24
3. Priznanja	33/2/17	452	46	0,40	0,04
S7				broju poveznica koje je potrebno izmijeniti	
1. Knjiga	10.ai	789	125	78,90	12,50
2. Knjiga	16.eps	1.254	130	78,38	8,13
S8				broju poveznica koje je potrebno izmijeniti	
1. Brošura	13	544	56	41,85	4,31
2. Časopis	65	2.409	295	37,06	4,55

Kod standardnog izvršenja zadatka brzina izrade raste povećanjem opsega dokumenta. Kod skriptnog izvršenja zadatka povećanjem opsega dokumenta do određene granice dobivamo brže rješavanje zadataka po karakteristikama zadatka koji se rješava. Navedeno je prikazano u Tablici 36. Velikim povećanjem opsega dokumenta može se smanjiti brzina

skriptnog izvršavanja zbog povećanog pristupa DOM-u što produžuje vrijeme izvršavanja (npr. skripta S3). Navedeno će ovisiti o veličini dokumenta i radnoj memoriji koju skripta ima na raspolaganju pri izvršavanju zadatka. Brzina izvođenja skripte može biti smanjena i ukoliko je korisnik uključen u zadatke izvršavanja skripte (npr. skripta S5).

Tablica 37. Poredak dobivenih prosječnih vremenskih ušteda skriptnog izvršavanja zadatka te prosječno standardno vrijeme trajanja izvršenja zadatka

	Testirani zadatci	Prosječna ušteda vremena pri skriptnom izvršavanju zadatka	Prosječno standardno vrijeme izvršavanja u sekundama	Utjecaj korisnika na vrijeme skriptnog izvršavanja
S3	Uređivanje tablice u periodičkoj publikaciji	95 %	1364	NE
S4	Numeracija i montaža kartica	94 %	981	DA
S6	Izrada jednostavnih montaža	90 %	334	DA
S8	Automatska obrada fotografija	89 %	1477	NE
S7	ePub dokument sa SVG datotekama	87 %	1022	DA
S1	Brisanje neželjenih bjelina u tekstu	86 %	88	NE
S5	Izrada montaže i personalizacije iz XML datoteke	81 %	720	DA
S2	Zadržavanje uvezenih stilova	78 %	81	NE

U Tablici 37. skripte su poredane prema vremenskim uštedama. Najmanja je vremenska ušteda postignuta kod skripte S2 koja služi za zadržavanje uvezenih stilova. Kod standardnog izvođenja zadataka zadržavanja uvezenih stilova postoji djelomična automatizacija u rješavanju upotrebom *Find/Change* izbornika. Ista opcija upotrijebljena je i kod skripte S1 za brisanje neželjenih bjelina pa je i u navedenoj skripti postignuta nešto manja vremenska ušteda. Skripta S5, koja služi za izradu montaže i personalizacije iz XML datoteke, također je bila optimizirana s nešto manjim postotkom skraćivanja vremena. Kao kod skripti S2 i S1, i ovdje kod standardnog izvođenja zadatka postoji mogućnost u programu za djelomično automatiziran rad s XML datotekama pomoću paleta *Tags* i *Structure* te *Autoflow* opcije tekstualnih okvira. Kod skripte S5 dobiveni su nešto lošiji rezultati vremenske optimizacije i zbog utjecaja vremena korisnika koji određuje način dodavanja XML elemenata.

Također je vidljivo iz Tablice 37. da se vrlo dobra optimizacija vremena može postići kod dugotrajnih zadataka i zadataka za koje nije predviđeno odrađivanje u programu za koji je

pisana skripta. Za takve zadatke ne postoje automatizacije u programu, već je sve korake potrebno odraditi standardnom izradom. Kod skripte S3 potrebno je postaviti velik broj stilova i formatiranja na tekst i tablicu. Jedini je način da se izvrši navedeni zadatak bez upotrebe skriptiranja, da se uređuje pojedinačno svaki redak tabele što iziskuje mnogo vremena i rada korisnika programa. Kod skripti S4 i S6 također ne postoji mogućnost automatizacije pri izradi montaže, osim uz pomoć skriptiranja.

Skripte S7 i S8, koje omogućuju automatsku obradu fotografija i izradu SVG datoteka, dale su dobre rezultate. Iz njih se može zaključiti da je moguće pisati skripte koje će optimizirati procese rada u više programa grafičke pripreme, a koje uvelike skraćuju dugotrajna vremena izrade.

Skripta S3 dala je najveću uštedu vremena. Navedena skripta testirana je i u izradi stvarne publikacije, ne samo za potrebe ovog rada. Testirana periodička publikacija bila je Službeni glasnik Međimurske županije. Kako se radi o periodičkoj publikaciji, lako se može izvesti izračun koji ukazuje na znatnu uštedu resursa. U svakom objavljenom Glasniku nalazi se više tablica proračuna koje je moguće pripremiti za objavu putem skripti. Testiranjem i upotrebom skripte za izradu publikacije dobivena je vremenska ušteda od skoro 9 sati za jedan broj periodičke publikacije [4]. Ukoliko bi se pretpostavilo da je to prosječna ušteda po jednom Glasniku (broj tablica proračuna razlikuje se u izdanjima), okvirno bi se moglo izračunati da se na 20 glasnika, koliko ih prosječno izlazi godišnje, dobiva ušteda od oko 22 radna dana jednog zaposlenika. Navedena ušteda ukazuje na velik potencijal skripti pri optimizaciji vremena trajanja digitalne grafičke pripreme.

Kod vremenski kraćih skripti ukupno vrijeme uštede nije toliko impresivno kao u navedenom slučaju. Upotrebom skripti koje mijenjaju kraće zadatke digitalne pripreme, dobivaju se poboljšanja u proizvodnom procesu zbog automatizacije, vjerojatnosti optimalnog rješavanja zadatka s minimalnim brojem koraka te pouzdanosti izrade pripreme koja ima manju vjerojatnost da će sadržavati neispravljene greške.

Za skriptno izvršenje zadatka moguće je zaključiti da se manja vremenska optimizacija u postotku može očekivati kod zadataka koji su predviđeni za standardnu izradu u programu za koji se piše skripta. Uobičajeno je da se za navedene zadatke putem izbornika programa mogu odabrati metode djelomične automatizacije zadataka. Najbolje će se optimizirati zadatci koji nisu podržani za izradu u programu i za koje ne postoji automatizacija putem izbornika.

Povećanjem veličine dokumenta koji se uređuje, uobičajeno se ukupna vremenska ušteda povećava, ali se ušteda vremena u postotcima prema standardnom izvršavanju

smanjuje. Pri kraćim vremenima standardnog izvršavanja zadatka ukupna je vremenska ušteda manje značajna, ali u postotku i dalje izrazito visoka. Najznačajnija se vremenska ušteda može dobiti skriptnim izvođenjem dugotrajnih zadataka i zadataka koji zahtijevaju mnogo slijednih koraka.

Kod skriptnog je izvođenja zadatka utjecaj korisnika na vrijeme izvođenja malen. Na skriptno vrijeme izvršavanja utjecat će zadatci koji se izvode te način na koji je skripta napisana, kao i računalni sustav u kojemu se skripte izvršavaju. Na skriptno izvođenje zadatka može utjecati i opseg dokumenta koji se uređuje. Kod standardnog rješavanja zadatka najveći je utjecaj korisnika i zadatka koji se izvršava.

6.2. Automatizacija izvršavanja zadataka grafičke pripreme

Dobivena automatizacija u svim je skriptama izuzetno velika i kreće se od 83 % do 100 % kao što je vidljivo na Slici 93. i u Tablici 38. Prosječno je postignuta automatizacija, tj. smanjenje broja potrebnih koraka u izvršavanju zadatka u iznosu od 94 %. Iz Tablice je vidljivo da, suprotno od optimizacije vremena, povećanjem obima zadatka koji se izvršava dobivamo veću automatizaciju procesa skriptnim izvođenjem zadatka. Na Slici 93. vidljivo je da porastom ukupne automatizacije procesa raste i ukupan postotak vremenske uštede te da skripte uobičajeno daju veći postotak automatizacije od postotka vremenske uštede.

Tablica 39. prikazuje prosječne uštede potrebnog broja koraka u izvršenju zadatka te prosječne vrijednosti potrebnih klikova mišem i pritiska tipkovnice za sva mjerenja u pojedinom zadatku standardnog i skriptnog izvršavanja. Iz dobivenih je rezultata vidljivo da je najveća automatizacija postignuta kod skripti S3 i S8 gdje imamo velik broj koraka pri standardnom rješavanju zadatka, a pri skriptnom izvršavanju nema varijacija u broju koraka. Skripte su izrađene tako da odrađuju zadatak bez intervencija od strane korisnika. Zadatci koji se obavljaju uvijek su isti pa nema potrebe sudjelovanja korisnika u rješavanju zadatka. Kod standardnog izvršenja navedenih zadataka povećanjem opsega dokumenta povećava se i opseg zadataka koje korisnik mora obaviti. Tablice s više redaka ili prijelomi s više fotografija kod standardnog izvršavanja zahtijevaju više koraka.

Broj potrebnih koraka kod standardnog izvođenja zadatka može ovisiti o dokumentu koji se prelama, načinu izvođenja zadatka i osobi koja zadatke izvršava. Ukoliko bi zadatke izvršavale osobe koje nisu stručnjaci grafičke pripreme standardnim izvršenjem zadatka, dobili bi mnogo lošije rezultate. Prednost je skriptnog izvršavanja zadatka da ne ovisi o znanjima i iskustvima korisnika.

Kod skripte S4, koja zamjenjuje zadatke numeracije i montaže kartica, varijacije u broju koraka događaju se i pri skriptnom rješavanju zadatka. Kod navedene skripte, zbog većeg broja varijabli koje se mogu mijenjati, dolazi i do varijacija u potrebnom broju koraka. Skripte koje ovise i o korisniku imaju varijacije u broju potrebnih koraka za izvršenje zadatka. Varijacije se događaju pri pozicioniranju okvira na kartici, podešavanju postavki numeracije i montaže. Ostali dijelovi zadatka su automatizirani. Zbog velikog broja potrebnih koraka pri standardnom izvršenju zadatka i za ovu se skriptu može zaključiti da je dobro optimizirana.

Proces je najmanje optimiziran u broju koraka kod skripte S5. Kod standardnog izvršavanja zadataka skripte S5 postoji automatizacija koja se može primijeniti. Ona će

smanjiti broj koraka standardnog izvršenja zadatka pa je ukupna skriptno postignuta automatizacija procesa manja.

Tablica 38. Dobivena automatizacija, tj. smanjenje potrebnog broja klikova i pritisaka tipki tipkovnice kod skriptnog izvršavanja zadatka u usporedbi sa standardnom izradom po testiranim skriptama

	Testirani zadatci	1.	2.	3.	4.	5.	Prosjek
S1	Brisanje neželjenih bjelina u tekstu	93 %	93 %	93 %	93 %		93 %
S2	Zadržavanje uvezenih stilova	91 %	91 %	92 %			91 %
S3	Uređivanje tablice u periodičkoj publikaciji	99 %	100 %	100 %			100 %
S4	Numeracija i montaža kartica	95 %	97 %	97 %	96 %	96 %	96 %
S5	Izrada montaže i personalizacije iz XML datoteke	85 %	83 %	87 %			85 %
S6	Izrada jednostavnih montaža	92 %	92 %	97 %			94 %
S7	ePub dokument sa SVG datotekama	89 %	93 %				91 %
S8	Automatska obrada fotografija	98 %	100 %				99 %
Prosjek:							94 %

Tablica 39. Poredak prosječnih dobivenih automatizacija, tj. smanjenja potrebnog broja klikova i pritisaka tipki tipkovnice

	Testirani zadatci	Prosjek	Standardno	Skriptno	Promjenom opsega zadatka povećava se potreban broj klikova i pritisaka tipkovnice kod zadatka koji se izvodi
S3	Uređivanje tablice u periodičkoj publikaciji	100 %	1724	4	standardno
S8	Automatska obrada fotografija	99 %	654	5	standardno
S4	Numeracija i montaža kartica	96 %	1134	42	standardno i skriptno
S6	Izrada jednostavnih montaža	94 %	242	14	standardno
S1	Brisanje neželjenih bjelina u tekstu	93 %	70	5	-
S2	Zadržavanje uvezenih stilova	91 %	46	4	-
S7	ePub dokument sa SVG datotekama	91 %	707	58	standardno
S5	Izrada montaže i personalizacije iz XML datoteke	85 %	328	50	standardno i skriptno

Kod skripti S5 i S7 do manje optimizacije dolazi i jer pri skriptnom izvršavanju zadatka postoje dijelovi koji nisu automatizirani, već te zadatke mora obaviti korisnik. Skripte

S2 i S1 daju manju automatizaciju jer kod standardnog izvršenja zadatka nije potreban velik broj koraka.

Manja automatizacija skriptnim tehnologijama može se očekivati kod zadataka grafičke pripreme koji kod standardnog izvođenja ne zahtijevaju velik broj koraka, kod skripti pri čijem je izvođenju uključen korisnik i za skriptne zadatke kod kojih u programu postoji ugrađena automatizacija putem izbornika. Najbolje se automatiziraju procesi digitalne grafičke pripreme gdje se skriptnim tehnologijama mijenjaju kompleksni zadatci s mnogo koraka i zadatci koji nisu predviđeni za obavljanje u programu za koji se piše skripta.

6.3. Pouzdanost optimalnog i ispravnog izvršenja zadatka

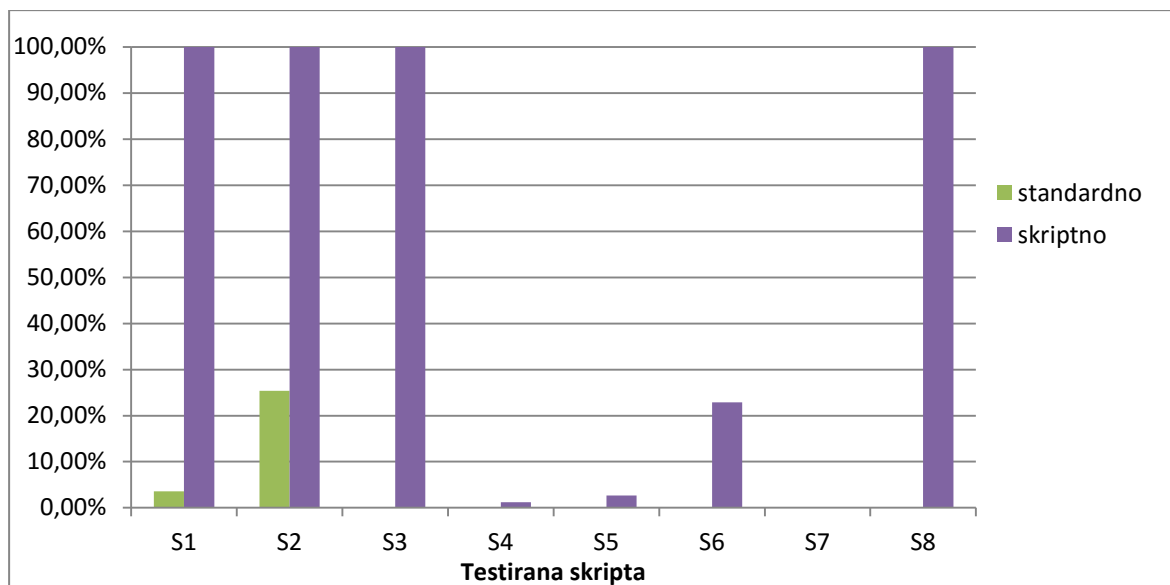
Rezultati vjerojatnosti optimalnog rješavanja zadataka s minimalnim brojem koraka za svaki od standardno ili skriptno izrađivanih zadataka prikazani su u Tablici 40. i na Slici 94. Iz dobivenih rezultata vidljivo je da se zadatci puno bolje mogu izvršavati upotrebom skriptnih tehnologija. Pri rješavanju zadatka pomoću skripte uobičajeno je potreban mnogo manji broj koraka za izvršenje pa je time i vjerojatnost da će korisnik optimalno rješavati zadatke mnogo veća.

Kako je vidljivo iz Tablice 39., najveću vjerojatnost da će se zadatak optimalno riješiti daju skripte koje ne zahtijevaju intervenciju korisnika (S1, S2, S3 i S8). To su skripte koje odrađuju zadatke uvijek na jednak način i automatiziraju učestale ponavljajuće zadatke grafičke pripreme. Manju vjerojatnost da će se zadatak optimalno riješiti imaju skripte kod kojih se očekuje intervencija korisnika. To su skripte čije izvršavanje ovisi o korisnikovim postavkama.

Kod standardnog izvršavanja zadatka mogućnost rješavanja s minimalnim brojem koraka je iznimno niska. Minimalan broj koraka moguće je postići jedino kod malog broja potrebnih koraka, kratkotrajnih i jednostavnih zadataka. U svim ostalim slučajevima broj koraka uvelike varira jer pri radu uobičajeno nastaju pogreške koje je potrebno ispraviti.

Tablica 40. Dobivene pouzdanosti optimalnog rješavanja zadatka standardnom i skriptnom izradom

	Testirani zadatci	standardno	skriptno
S1	Brisanje neželjenih bjelina u tekstu	3,56 %	100 %
S2	Zadržavanje uvezenih stilova	25,39 %	100 %
S3	Uređivanje tablice u periodičkoj publikaciji	0 %	100 %
S4	Numeracija i montaža kartica	0 %	1,20 %
S5	Izrada montaže i personalizacije iz XML datoteke	0 %	2,70 %
S6	Izrada jednostavnih montaža	0 %	22,90 %
S7	ePub dokument sa SVG datotekama	0 %	0,11 %
S8	Automatska obrada fotografija	0 %	100 %
	Prosjek:	4 %	53 %



Slika 94. Usporedba pouzdanosti optimalnog rješavanja zadatka standardnom i skriptnom izradom

Rezultati pokazuju da zadatci koji se izvršavaju u programima standardnim načinom izrade grafičke pripreme, ukoliko su kompleksni, dugotrajni, ponavljajući, zahtijevaju velik broj koraka te imaju otežanu vizualnu kontrolu gotove pripreme, obično imaju veliku vjerojatnost da će pri prijelomu nastati pogreška i da neće biti ostvareni s optimalnim brojem koraka. Složeni zadatci s mnogo različitih slijednih koraka imaju veću varijaciju u izvođenju od jednostavnih ponavljajućih zadataka.

Poseban problem pri izradi grafičke pripreme predstavljaju ponavljajući zadatci koji su dugotrajni, a ne postoji jednostavna vizualna kontrola odrađenog zadatka. Olakšana vizualna kontrola gotove pripreme smanjuje vjerojatnost nastanka pogreške koja ostaje neispravljena. Standardno izvršavanje zadataka kod skripti S4 i S8 predstavlja zadatke koji su dugotrajni i ponavljajući, ali imaju olakšanu vizualnu kontrolu gotove grafičke pripreme i iz tog je razloga mnogo manja vjerojatnost pogreške koja ostaje neispravljena. Kod standardnog je izvršavanja zadatka skripte S4 za izradu numeracije i montaže kartica pri završetku pripreme moguće provjeriti prvu i posljednju stranicu i na taj način provjeriti jesu li svi brojevi dobro pozicionirani u povezanim okvirima. Provjera standardnog izvršenja zadatka skripte S8 koja obrađuje fotografije može se obaviti preko *Links* palete gdje je moguće vidjeti osnovne informacije o fotografijama koje se koriste u dokumentu.

Većina zadataka koji su se testirali u ovom radu jednostavni su i relativno kratkotrajni ili kompleksni, ali s jednostavnom vizualnom kontrolom pripreme. Zbog navedenih karakteristika zadataka, u konačnoj pripremi, osim kod skripte S3 za uređenje tablica u

periodičkoj publikaciji, nisu nastale pogreške koje su ostale neispravljene. Upravo skripta S3 predstavlja zadatak koji je dugotrajan, s mnogo slijednih koraka, ponavljajući i s otežanom vizualnom kontrolom. Pri standardnom rješavanju zadatka skripte S3 u prosjeku su se dogodile po 3 pogreške u prijelomu koje su ispravljene nakon korekture. Samo je jedan od 15 standardno rađenih prijeloma dao u potpunosti ispravno prelomljen dokument, tj. početno je 93 % dokumenta bilo neispravno prelomljeno što je iznimno visok postotak pogrešaka. Greške je moguće pronaći jedino naknadnom provjerom svakog retka tabele što iziskuje značajno dodatno vrijeme za kontrolu pripreme.

Standardnom izradom jednostavnijih zadataka s manjim brojem koraka obično se dobiva točnija priprema te je varijacija u broju koraka manja. Ispravnost grafičke pripreme ovisi o zadatku koji se izvršava te o mogućnosti vizualne kontrole izrađenih zadataka. Jednostavnost kontrole gotove pripreme utječe na ispravnost. Varijacije u broju potrebnih koraka za ostvarenje pojedinog zadatka ovise o kompleksnosti i vrsti zadatka te stručnosti osobe koja ih izvršava.

Skriptni rezultati ukazuju na činjenicu da je skriptno rješavanje zadatka mnogo pouzdanije i veća je vjerojatnost ostvarivanja zadataka s minimalnim brojem potrebnih koraka. Kod skriptnog načina izvođenja zadatka kasnija korektura nije pronašla nijednu neispravljenu pogrešku. Na točnost skriptnog izvršavanja ne utječe vrsta zadatka koji se izvršava. Ukoliko su skripte dobro testirane, vjerojatnost pogreške gotovo da ne postoji. Pogreške mogu nastati pri interakciji korisnika sa sučeljem skripte ili pri drugim zadacima koje mora obaviti korisnik.

Skriptnim izvođenjem jednostavnih ili kompleksnih zadataka možemo dobiti visoku vjerojatnost ostvarenja zadatka s minimalnim brojem koraka. Pri skriptnom izvođenju zadatka broj potrebnih koraka za izvršavanje obično ne ovisi o kompleksnosti zadatka, već o načinu izvođenja skripte. Najlošije rezultate daju skripte koje ovise o korisniku. Tamo gdje je moguće isključiti korisnika iz izvršavanja zadatka, vjerojatnost minimalnog broja potrebnih koraka gotovo je osigurana. Najlošiji rezultati skripti pri vjerojatnosti ostvarenja zadatka s minimalnim brojem koraka dobiveni su za skripte S7, S4 i S5. Sve navedene skripte ovise o korisniku. Kod skripte S7 korisnik je zadužen za otpakiravanje gotovog ePub dokumenta i za ukazivanje na mjesto gdje se nalaze dijelovi ePub dokumenta koje je potrebno izmijeniti sa SVG datotekama. Sama skripta sačinjena je zapravo od 2 skripte što također smanjuje automatizaciju i vjerojatnost minimalnog broja koraka pri izvršenju skripte. Kod skripti S4 i S5 korisnik pozicionira okvire na kartici i zadužen je za ispravne odabire u menijima skripte.

Skripta S5 također dodatno zahtijeva korisnikovo povezivanje XML elemenata s okvirima u koje će se naknadno ubaciti.

Poglavljima 6.1., 6.2. i 6.3. potvrđena je hipoteza koja tvrdi da je moguće odrediti značajke radnih procesa pomoću kojih će se kvantificirati uštede i prednosti koje se dobivaju upotrebom skriptnih tehnologija. Za sve nove modele vrednovana je učinkovitost pomoću kvantificiranja postignutih ubrzanja, automatizacije i povećanja pouzdanosti radnih procesa u grafičkoj pripremi.

6.4. Optimalno rješavanje zahtjevnih zadataka koji su teški za ostvarivanje

Svim izrađenim skriptama, a osobito skriptama S9 za izradu barkoda iz varijabilnih podataka i S10 za obilježavanje ispravaka, dokazana je mogućnost optimalnog rješavanja zahtjevnih zadataka. Novi skriptni modeli S9 i S10 integrirani sa standardnim programom omogućuju optimalno rješavanje zahtjevnih zadataka grafičke pripreme koji su inače teški i neisplativi za ostvarenje. Navedeni skriptni modeli samo su primjer novih mogućnosti koje se mogu dodati za olakšavanje svakodnevnih poslova grafičke pripreme. Glavna im je prednost olakšavanje i automatizacija rada bez potrebe za ulaganjem u skupe komercijalne alate te dodavanje novih mogućnosti koje su inače nedostupne u programu.

Skriptom S9 olakšava se zadatak koji je inače težak za ostvarenje u standardnom programu. Kako programom nije predviđena mogućnost dodavanja barkoda na stranice, ovakav se zadatak omogućuje skriptno. Izrada bez upotrebe skripte bila bi mnogo kompleksnija i iznimno teška za ostvarenje zbog otežane vizualne kontrole dodanih barkodova. Izradom skripte omogućuje se nova mogućnost u programu te nije potrebno ulagati u komercijalne alate.

Skriptom S10 također je dodana mogućnost koja nije postojala u programu. U standardnom testiranom programu nije postojala mogućnost obilježavanja ispravaka za slanje naručiocu na odobrenje elektroničkim putem. Navedena mogućnost uvelike je nedostajala pri radu u programu te se odobravanje ispravaka nije moglo izvoditi digitalno. Izrađena skripta donijela je velike prednosti i uštede pri postupcima korekture i obilježavanja ispravaka te odobrenja od strane naručioca.

Kod skripte S2 također je izrađena mogućnost koja je teška za ostvarenje. Ta dodana mogućnost je traženje velikog broja kombinacija znakovnih stilova koji se mogu pojaviti u dokumentu. Traženje 144 kombinacije znakovnih stilova standardnim načinom izvođenja bilo bi neisplativo i teško ostvarivo. Skripta S2, koja traži sve znakovne stilove i njihove kombinacije, omogućuje precizniju pripremu te daje veću pouzdanost izradi kvalitetne pripreme u kojoj neće doći do neželjenih izmjena u stilu teksta koji se prelama.

Izrađene skripte dokazuju da skriptne tehnologije omogućuju nove modele rada u postojećim standardnim programima grafičke pripreme. Nove mogućnosti olakšavaju rad i mogu optimizirati proizvodne procese. Skriptiranjem se mogu ostvarivati zadatci za koje ne postoji optimalan standardan način izvršavanja zadatka, zadatci koji su zahtjevni, teški ili neisplativi za izradu u standardnim programima grafičke pripreme.

Izrađenim skriptama potvrđena je druga hipoteza rada koja tvrdi da je moguće dodavanjem novih skriptnih modela u standardne programe grafičke pripreme optimalno rješavati zahtjevne zadatke koji su inače teški za ostvarivanje

6.5. Kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju korištenjem skriptnih tehnologija

Analiza proizvodnih procesa grafičke pripreme preduvjet je za identificiranje procesa koji se mogu optimizirati upotrebom skriptnih tehnologija. Kako i samo pisanje skripti zahtijeva vrijeme za izradu i testiranje, teško je bez analize odrediti procese poželjne za automatizaciju. Skripte koje se izrađuju mogu optimizirati zadatke generalne i specifične namjene pa ih je potrebno prilagoditi stvarnim proizvodnim procesima kako bi se izrađivale one koje će najbolje optimizirati radne procese unutar analiziranog poduzeća.

Iz izvedene analize proizvodnih procesa u ovom radu, izrađenih i testiranih skripti vidljivo je da skripte specifične namjene daju najviši stupanj automatizacije i optimizacije vremena izrade (skripta S3), ali samo ukoliko se primjenjuju na ponavljajuće zadatke grafičke pripreme. Skripte generalne namjene primjenjuju se na sve procese rada (skripte S1 i S2) ili samo na određene tipove zadataka (S4, S5, S6, S7, S8, S9, S10).

Analizom, testiranjem i primjenom skriptnih tehnologija određeni su kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju:

Osnovni kriterij:

1. Kriterij izvedivosti skripte u više radnih procesa.

Dodatni kriteriji:

1. Proces je vremenski zahtjevan i kompleksan, s velikim brojem zadataka koje je potrebno obaviti u zadanom redoslijedu, a izrađena automatizacija može se primjenjivati i na druge procese (S4, S7, S8).
2. Proces je sklon pogreškama zbog otežane vizualne kontrole gotove pripreme (S3, S9).
3. Proces je jednostavan, ali ponavljajući, dugotrajan i zamoran kad se odrađuje standardnom izradom. Materijal koji se prelama sastoji se od mnogo stranica ili elemenata koje je potrebno ručno uređivati (S3, S7, S8, S10).
4. Materijal se objavljuje s podacima, tj. sadržajem ili dijelovima sadržaja koji dolaze iz baze podataka, XML dokumenta ili je izrađen pomoću skripte. Posao se odnosi na personalizaciju publikacije i varijabilni tisak (S4, S5, S9).
5. Novi model rada koji je moguće implementirati u programe za DTP kako bi se izbjegla dodatna ulaganja u komercijalne alate (S7, S9, S10).

6. Ostvarivanje zadataka koje bi bilo teško ili nemoguće ostvariti ostalim sredstvima (S2, S9, S10).
7. Skripta automatizira kratkotrajne, ali svakodnevne zadatke i česte postupke grafičke pripreme, općenita je i primjenjiva na mnogo procesa (S1, S2).
8. Poslovi koji uključuju izradu ponavljajućih zadataka u više programa grafičke pripreme (S7, S8).
9. Prilagodba standardnih programa vlastitim radnim navikama i poslovima (S4, S6, S7, S8).

Pri izradi skripte potrebno je osigurati osnovni kriterij izvedivosti skripte u više radnih procesa jer je to glavni čimbenik koji omogućuje postizanje optimizacije digitalne grafičke pripreme. Ukoliko izradimo skriptu koja je primjenjiva samo u rijetkim i specifičnim slučajevima, stvarna ušteda vremena i postignuta automatizacija često će biti zanemarive u usporedbi s vremenom utrošenim za pisanje skripte.

Nakon što su identificirani potencijalni procesi koje želimo unaprijediti, prije pisanja skripte potrebno je izvesti analizu izvedivosti pojedinog modela i odrediti nivo automatizacije. Proces se odabiru tako da napisane skripte budu općenitije i primjenjive u većem obimu procesa grafičke pripreme. Ukoliko se izrađuju složenije skripte koje izvršavaju više zadataka, to se može postići uvođenjem promjenjivih parametara (S4, S5, S6) koje određuje korisnik skripte.

Skripte koje uključuju korisnika u izvršavanje zadatka nešto se lošije automatiziraju, smanjena je vjerojatnost ostvarenja zadatka s minimalnim brojem koraka i obično je potrebno dulje vrijeme izvršavanja. Usprkos navedenim nedostacima, prednosti koje se dobivaju znatno su značajnije. Bolje je jednom skriptom rješavati više sličnih zadataka, nego se ograničiti samo na specifične zadatke koji se rijetko događaju. Dinamičke skripte, koje ovise o zadatku koji se trenutno obavlja, moguće je ostvariti i s promjenjivim varijablama koje se određuju iz dokumenta koji se uređuje (S3, S5) pa je na taj način u određenim slučajevima moguće isključiti nedostatke koje donosi uključivanje korisnika u izvršenje zadatka.

Ukoliko biramo procese koji zadovoljavaju navedene kriterije, vrijeme koje je potrebno za pisanje skripte može se zanemariti. Skripte je poželjno pisati tako da se mogu primjenjivati na više sličnih poslova grafičke pripreme i na česte situacije koje se pojavljuju pri prijelomu dokumenta. Vrijeme izrade skripte moguće je u potpunosti izbaciti iz razmatranja isplativosti ukoliko se koriste već gotove skripte. Gotove skripte mogu se naći na brojnim web lokacijama za vrlo različite zadatke u programu.

Izrađena analiza segmenata proizvodnih procesa grafičke pripreme koji bi se mogli upotrebom skriptnih jezika pod određenim uvjetima optimalno ubrzati s povećanjem pouzdanosti i isplativosti, zadovoljava traženi izvorni znanstveni doprinos disertacije. Definirani su i kriteriji za optimalnu upotrebu skriptnih tehnologija u standardnim programima i metodama rada.

Iz dosad navedenog potvrđena je prva hipoteza koja tvrdi da je moguće grafičku pripremu temeljem analize segmenata proizvodnih procesa grafičke pripreme, upotrebom skriptnih jezika pod određenim uvjetima optimalno ubrzati s povećanjem pouzdanosti i isplativosti.

Određeni kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju upotrebom skriptnih tehnologija potvrđuju hipotezu koja tvrdi da je moguće zamjenom samo određenih radnih procesa skriptnim datotekama automatizirati procese grafičke pripreme jer postoje zadatci grafičke pripreme za koje nije optimalna upotreba skriptnih tehnologija. Zadatci za koje nije optimalno korištenje skriptnih tehnologija navedeni su u poglavlju 6.8. Zadatci grafičke pripreme za koje nije optimalno korištenje skriptnih tehnologija.

6.6. Uštede i prednosti koje se dobivaju korištenjem skriptnih tehnologija

Istraživanje i ostvareni rezultati pokazuju kako je pomoću skriptiranja moguće ostvarivati zadatke koji su dugotrajni i iznimno zahtjevni za ostvarenje što je dokazano izrađenim skriptama. Pomoću skripte smanjuje se vrijeme izrade, automatizira proces i povećava pouzdanost. Pouzdanost se povećava ovisno o smanjenju interakcije korisnika s grafičkim sučeljem. Korištenjem skriptnih tehnologija moguće je ostvarivati zadatke koji se inače ne mogu ostvarivati u standardnim programima grafičke pripreme.

Prilikom testiranja izrađenih skripti određene su i dokazane uštede i prednosti korištenja skriptnih tehnologija u ostvarenju zadataka digitalne grafičke pripreme.

Svim skriptama dokazane su sljedeće prednosti korištenja skriptnih tehnologija:

- Nijedna druga mogućnost dostupnih alata u standardnim programima digitalne grafičke pripreme ne može toliko uštedjeti rad, vrijeme i novac.
- Korištenjem skriptnih tehnologija olakšava se izrada digitalne grafičke pripreme.
- Novi modeli rada koji koriste skriptne tehnologije jednostavniji su u primjeni.
- Proces digitalne grafičke pripreme moguće je unaprijediti skraćivanjem vremena izrade.
- Korištenjem skriptnih tehnologija povećava se stupanj automatizacije izrade digitalne grafičke pripreme.
- Korištenjem skriptnih tehnologija povećava se pouzdanost sustava smanjenjem mogućnosti nastanka pogrešaka koje ostaju neispravljene.
- Korištenjem skriptnih tehnologija povećava se vjerojatnost ostvarenja zadatka s minimalnim brojem koraka.
- Vrijeme izvršavanja, potreban broj koraka i pouzdanost ne moraju ovisiti o stručnosti osobe koja izvršava zadatak.

Pojedinačne skripte dokazuju sljedeće:

- Korištenjem skriptiranja moguće je olakšati i ubrzati zadatke koji se standardno izvršavaju u programu (S1, S2, S3, S4, S5, S6, S7, S8).
- Mogućnost optimizacije učestalih zadataka grafičke pripreme (S1, S2, S8).
- Korištenjem skriptiranja moguće je olakšati i ubrzati rad s XML podacima pri izradi grafičke pripreme iako sam program nudi ugrađenu automatizaciju za rad s XML podacima (S5).

- Unutar jedne skripte moguće je korištenjem skriptiranja optimizirati rad u više standardnih programa digitalne grafičke pripreme (S7 i S8).
- Zamjenom nestandardnih radnih procesa skriptnim tehnologijama moguće je optimizirati procese grafičke pripreme i rješavati zadatke koji su teški za ostvarivanje (S4, S6, S7).
- Izrađena skripta može omogućiti nove mogućnosti i modele rada kod zadataka za koje ne postoji optimalan standardan način izvršenja, koji su zahtjevni, teški ili neisplativi za ostvarenje standardnom izradom (S2, S9, S10).
- Mogućnost izrade skripte koja omogućuje unošenje varijabilnih podataka kako bi se mogla primijeniti na različite proizvodne procese (S4, S5, S6, S9).
- Jednom napisana skripta može se preuređivati i nadograđivati za eventualne promjene koje se uključuju u skriptu, kako bi se proširili zadatci koje skripta ostvaruje (S4, S5, S6).
- Mogućnost primjene i prilagodbe skripti koje su kreirali drugi.

6.7. Optimalno korištenje skriptnih tehnologija u standardnim programima i metodama rada

ExtendScript skripte koje se izvršavaju u Adobe programima mogu biti dugotrajne. Za vrijeme trajanja skripte korisnik programa ne može raditi u njemu. Upravo kod dugotrajnih skripti poželjno je izvršiti optimizaciju skripte kako bi se skratilo vrijeme izvršavanja i omogućio nastavak s radom u programu. Optimizacijom koda se, osim vremena trajanja skripte, može optimizirati i veličina datoteke koja se u konačnici dobiva.

Za testiranje optimizacije napisanog koda odabrana je skripta S4 koja je imala dugo vrijeme izvođenja. Bilo je pretpostavljeno, temeljem teorijskih pretpostavki, da je optimizacijom koda moguće izraditi skriptu koja se izvršava kraće vrijeme, što je i dokazano. Testiranjem su utvrđeni čimbenici koji utječu na vrijeme izvršavanja ExtendScript koda.

Iz dobivenih rezultata zaključuje se da optimizacija funkcija koje ne pristupaju DOM-u ne dovodi do značajnijeg ubrzanja skripte. Ubrzanje izvršavanja skriptnog koda moguće je ostvariti optimiziranjem dijelova koda koji pristupaju DOM-u dokumenta. Primjenom optimizacije može se postići velika ušteda u vremenu izvršavanja skripte (oko 70 % u testiranom primjeru). Veću uštedu u ukupnom vremenu izvršavanja moguće je postići kod dugotrajnijih skripti.

Testiranjem su dobiveni zaključci i izrađene sljedeće preporuke za primjenu pri pisanju i optimizaciji skripti:

- Vrijeme izvršavanja skripte koja koristi „*UndoModes*“ opciju „*doScript*“ metode može se višestruko produžiti pa je obavezno testiranje prije upotrebe navedene opcije.
- Pri radu s mnogo okvira na stranicama dokumenta poželjno je koristiti povezane okvire umjesto zasebnih okvira s tekstem. Na navedeni način osigurat će se bolje skriptno vrijeme i manja veličina dobivene datoteke.
- Preporučuje se rad s objektima kolekcije u svim slučajevima gdje je to moguće, umjesto rada sa zasebnim objektima (upotreba „*everyItem()*“ kako bi se postavila svojstva objekata u kolekciji).
- Broj pristupa DOM-u potrebno je čim više smanjiti (dulje vrijeme će se izvršavati kod koji ima mnogo pristupa DOM-u).
- Preferirati „*for*“ petlje s fiksnim uvjetima bez nepotrebnih pristupanja DOM-u.
- U slučaju upotrebe „*while*“ petlje preferirati petlju koja se izvršava smanjivanjem uvjeta.

- Sve varijable koje se koriste u kodu potrebno je postaviti na lokalne varijable kako bi se smanjio kontekst izvođenja i lanac dosega varijabli.
- Spremanje razriješenih referenci prema DOM-u u lokalne varijable i ponovno korištenje iz varijabli prema potrebi („*getElement()*“ metoda ili spremanje reference u varijablu).
- Kad se pristupa DOM-u, potrebno se pri pisanju koda pridržavati vremenske i prostorne lokalnosti; zadatci se trebaju izvršavati na objektima koji su blizu u DOM-u ili na objektima koji su nedavno selektirani. Zadatke na objektima potrebno je rješavati tako da se na jednom selektiranom objektu obave svi potrebni zadatci odjednom, umjesto izvršavanja skripte zadatak po zadatak na svim objektima (Slika 74.).

6.8. Zadaci grafičke pripreme za koje nije optimalno korištenje skriptnih tehnologija

Prilikom analize procesa utvrđeni su i procesi koje je nepoželjno, teško ili nemoguće optimizirati korištenjem skriptnih tehnologija. Pritom su određeni sljedeći kriteriji koji opisuju procese koji nisu podobni za optimizaciju skriptiranjem:

- Zadaci koji se izvršavaju samo jednom ili se rijetko ponavljaju.
- Optimizacija prijeloma cjelokupnih periodičkih publikacija u kojima nije unaprijed određeno formatiranje dokumenta, nego je ovisno o odlukama dizajnera. Periodičke publikacije često je dobro optimizirati djelomično.
- Cjelokupni prijelom neperiodičkih publikacija koje imaju specifičan dizajn.
- Obrada fotografija koja je ovisna o slici koja se obrađuje.
- Izrada dizajna koja je ovisna o odlukama grafičkog dizajnera.
- Određivanje oblikovanja teksta; određivanje stilova i pozicioniranje tekstualnih okvira.
- Raspoređivanje elemenata po stranicama na predloškom zadane pozicije.
- Unošenje ispravaka u tekst.
- Popravci dostavljenih materijala za tisak (dodavanje napusta, micanje sadržaja od linije reza, skaliranje dokumenta, brisanje neispravno postavljenih reznih oznaka)
- Automatizacija izrade složenih montaža ukoliko je na raspolaganju alat namijenjen za izradu istih. Pod složenim montažama podrazumijevaju se montaže poslova s velikim brojem stranica, montaža knjiga za različite vrste uveza, upotreba različitih postavki montažnih araka u istom poslu i sl. Za složene montaže, ukoliko je moguće, bolje je koristiti već dostupna standardna rješenja, uobičajena i testirana u grafičkoj pripremi kao npr. program Preps. U njemu se montaže složenih poslova brže odrađuju zbog prilagodbe posla samo navedenim zadacima i zbog programskih jezika koji se brže izvršavaju od skriptnih jezika. Kod skriptiranja takvi poslovi mogu dugo trajati.

Pri odabiru procesa za automatizaciju nikako se ne smije zanemariti vrijeme potrebno za pisanje skripte. To će vrijeme u početku, dok se ne upoznaju svi objekti i način pisanja skripti, dok se ne stekne određeno iskustvo, biti čak izrazito dugo. Pretpostavka i dalje ostaje da će se investicija uskoro isplatiti ukoliko se pišu skripte za poslove pobrojane pod

poglavljem 6.5. Kriteriji pri odabiru procesa digitalne grafičke pripreme za optimizaciju korištenjem skriptnih tehnologija.

7. ZAKLJUČAK

Na temelju prikupljenih i obrađenih podataka potvrđene su sve početno postavljene hipoteze te je zadovoljen cilj istraživanja. Izrađenim skriptama dokazano je da se korištenjem skriptnih jezika u procesima grafičke pripreme postižu znatna unaprjeđenja povećanjem brzine i pouzdanosti te smanjenjem potrebnog broja koraka za izvršenje zadatka.

Prva hipoteza koja ističe da je moguće grafičku pripremu temeljem analize segmenata proizvodnih procesa korištenjem skriptnih jezika pod određenim uvjetima optimalno ubrzati s povećanjem pouzdanosti i isplativosti, dokazana je svim izrađenim mjerenjima koja uspoređuju standardnu izradu i skriptno izvršavanje zadatka. Temeljem analize procesa digitalne grafičke pripreme određeni su dijelovi procesa za koje su izrađene skripte. Izrađene skripte dale su iznimno visoku prosječnu vremensku uštedu od 87%. Kod testiranih skripti postignut je i visok stupanj automatizacije od prosječno 94 % smanjenja potrebnog broja koraka pri izvršenju zadatka. Skriptno izvršenje zadatka pokazalo se pouzdanije jer ni u jednoj testiranoj skripti nisu nastale pogreške koje su ostale neispravljene. Pouzdanost optimalnog rješavanja zadatka s minimalnim brojem koraka pri izradi kod skriptnog izvršenja zadatka uobičajeno je mnogo veća u usporedbi sa standardnim izvršenjem zadatka. Vremenska ušteda, dobivena automatizacija i pouzdanost ovisit će o zadacima koje skripta mijenja, korisniku koji izvršava zadatak, načinu kako je napisana skripta te karakteristikama sustava na kojemu se skripta izvršava. Iz izvedenih rezultata dokazana je hipoteza koja tvrdi da je moguće odrediti značajke radnih procesa pomoću kojih će se kvantificirati uštede i prednosti koje se dobivaju upotrebom skriptnih tehnologija. Važno je napomenuti da standardno vrijeme izvršavanja te potreban broj koraka uvelike ovise o operateru koji radi u programu, njegovom iskustvu i znanju. Dobiveni rezultati u ovom radu izvedeni su mjerenjem rada iskusnih operatera, a opet mogu varirati više od 80 % (zadatak S6) pri broju potrebnih koraka i više od 60 % (zadatak S5) pri vremenskim mjerenjima.

Dobiveni rezultati dokazuju da je moguće koristiti standardne alate u grafičkoj pripremi kako bi se ostvarili zadatci koje je bez skriptiranja teško ili nemoguće ostvariti. Za ostvarenje takvih modela moguće je koristiti skriptiranje koje omogućuje veliku fleksibilnost i efikasnost u rješavanju zadataka. Izrađenim skriptama dokazana je i druga hipoteza koja tvrdi da je moguće dodavanjem novih skriptnih modela u standardne programe grafičke pripreme optimalno rješavati zahtjevnije zadatke koji su inače teški za ostvarivanje. Navedenu hipotezu najbolje potvrđuju skripte za obilježavanje ispravaka u tekstu te skripta za ubacivanje barkod crteža. Skriptiranje omogućuje ostvarivanje dodane vrijednosti u grafičkim proizvodima bez

ulaganja u specijalne alate, uvelike smanjujući vrijeme izrade procesnih zadataka. Navedeno je najbolje vidljivo kod zadataka koji rade s varijabilnim podacima. Upotrebom skriptiranja može se skratiti vrijeme izrade pripreme za tisak što je iznimno značajno za manja poduzeća koja ne mogu uložiti u profesionalne alate za izradu varijabilnih podataka, ili je to manji dio njihovih zadataka, pa je ulaganje u navedene alate neekonomično.

Kriteriji za odabir procesa koje je poželjno optimizirati upotrebom skriptnih tehnologija te kriteriji koji ističu procese koje je nepoželjno, teško ili nemoguće optimizirati potvrđuju hipotezu da je zamjenom samo određenih radnih procesa skriptnim datotekama moguće automatizirati procese grafičke pripreme jer postoje zadatci grafičke pripreme za koje nije optimalno korištenje skriptnih tehnologija. Takvi zadatci mogu se vidjeti i u izrađenim skriptama gdje se uključuje operater u izvršenje zadatka za dijelove koje je teško skriptno izvršavati.

Izvršena mjerenja optimizacije samih skripti ukazuju na potrebu optimizacije skriptnog koda radi bržeg izvršavanja. Primjenom optimizacije skriptnog koda može se postići velika ušteda u vremenu izvršavanja skripte (oko 70 % u testiranom primjeru). Veću uštedu u ukupnom vremenu izvršavanja moguće je postići kod dugotrajnijih skripti pa je iste poželjno optimizirati. U radu su navedene testirane preporuke za primjenu pri pisanju i optimizaciji skripti čime su se definirali kriteriji za optimalno korištenje skriptnih tehnologija u standardnim programima i metodama rada.

Pri upotrebi skripti potrebno je obratiti pažnju na vrijeme potrebno za izradu, testiranje i korištenje skripte u usporedbi s vremenom koje bi se inače utrošilo na prijelom. Iz ovakve usporedbe može se zaključiti da je skriptiranje isplativo jedino u slučajevima kada imamo poslove koji se više puta ponavljaju ili kada skripte možemo više puta koristiti za različite zadatke. Navedeno ne vrijedi ukoliko koristimo gotove, provjerene skripte drugih korisnika. Tada se dobiva ušteda bez trošenja vremena na izradu i testiranje skripti.

Trošak uvođenja metoda skriptiranja čestih radnji je vrijeme koje je potrebno utrošiti na edukaciju, što ovisi o samom programeru/grafičkom dizajneru, njegovom predznanju iz tog područja, poznavanju DTP programa i objektnog modela. Iz dobivenih rezultata vidljivo je da se investicija na kraju može itekako dugoročno isplatiti. Vremenske uštede, postignuta automatizacija i pouzdanost prikazane u ovom radu upućuju na to da je skriptiranje dugoročno isplativa investicija, ali ovisna o samom korisniku i njegovoj upotrebi ovog alata.

LITERATURA

- [1] V. Žiljak, K. Pap, D. Agić, I. Žiljak, Modelling and simulation of integration of web system, digital and conventional printing, *29th International Research Conference of IARIGAI*, Lake of Lucerne, Switzerland, 2012.
- [2] C. Tuijn, Recent trends in print portals and Web2Print applications, *Proceedings of SPIE - The International Society for Optical Engineering*, svez. 7241, 2009.
- [3] D. P. Bianco, answers.com, Quark Inc., 2011. Dostupno na: <http://www.answers.com/topic/quark-inc>. (Pokušaj pristupa 11. 04.2013.).
- [4] S. Brekalo, Ž. Knok, N. Breslauer, Optimizacija procesa prijeloma publikacija korištenjem skriptiranja, u *Međunarodni znanstveni skup Tiskarstvo & Design*, Tuheljske Toplice, 2013.
- [5] Adobe Systems Incorporated, Scripting in InDesign, Adobe Systems Incorporated, Dostupno na: <http://helpx.adobe.com/indesign/using/scripting.html>. (Pokušaj pristupa 17. 05. 2013.).
- [6] Y. Zhao, W. Sun, Practice of imposition and illustrator variable data plate making with barcode, *17th IAPRI World Conference on Packaging, Tianjin, China*, pp. 251-253, 2010.
- [7] X. Lin, Active layout engine: Algorithms and applications in variable data printing, *Computer-Aided Design*, svez. 1, br. 38, pp. 444-456, 2006.
- [8] A. Heydon, G. Nelson, The Juno-2 constraint-based drawing editor, Palo Alto, California, USA: Technical Report 131a, Digital Systems Research Center, 1994.
- [9] V. Kumar, Algorithms for Constraint-Satisfaction Problems: A Survey, *AI Magazine*, svez. 1, br. 13, pp. 32-44, 1992.
- [10] C. Jacobs, W. Li, E. Schrier, D. Barger, D. Salein, Adaptive grid-based document layout, *ACM Trans Graph*, svez. 3, br. 22, pp. 838-847, 2003.
- [11] L. Purvis, S. Harrington, B. O'Sullivan, E.C. Freuder, Creating personalized documents: an optimization approach, *ACM symposium on Document engineering*, pp. 68-77, 2003.
- [12] A. Konaka, D. W. Coitb, A. E. Smithc, Multi-objective optimization using genetic algorithms: A tutorial, Elsevier Ltd., 2005.

- [13] S. Lok, S. Feiner, A Survey of Automated Layout Techniques for Information Presentations, Proceedings of the Smart Graphics Symposium, Hawthorne, NY, USA, 2001. Dostupno na: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.8736&rep=rep1&type=pdf>. (Pokušaj pristupa 13.09.2013.).
- [14] J. Geigel, A. Loui, Using Genetic Algorithms for Album Page Layouts, *IEEE MultiMedia*, svez. 4, br. 10, pp. 16-26, 2003.
- [15] R. Johari, J. Marks, A. Partovi, S. Shieber, Automatic yellow-pages pagination and layout, *Journal of Heuristics*, svez. 2, br. 4, pp. 16-26, 2003.
- [16] A. Berkner, E. L. Schwartz, SmartNails: Display- and image-dependent thumbnails, *SPIE conference on document recognition and retrieval XI, San Jose*, br. 54-65, 2004.
- [17] J. Geigel, A. Loui, Using Genetic Algorithms for Album Page Layouts, *IEEE MultiMedia*, svez. 4, br. 10, pp. 16-26, 2003.
- [18] B. Atkins, Adaptive photo collection page layout, *International conference on image processing, Singapore*, pp. 2897-2900, 2004.
- [19] E. Goldenberg, Automatic layout of variable-content print data, MSc Dissertation, School of Cognitive & Computing Sciences, University of Sussex, Brighton, UK, 2002. Dostupno na: <http://www.hpl.hp.com/techreports/2002/HPL-2002-286.pdf> (13.09.2013.).
- [20] K. J. Grahm, Efficient production of uniform layout, *Proceedings of the 1st international IARIGAI student conference on print and media technology, Chemnitz, Germany*, pp. 144-148, 2005.
- [21] K.J. Grahm., *Catalogue Production Automation – Case Studies*, Stocholm, Sweden: Master's Thesis in Publishing Technology at the School of Computer Science and Engineering, Royal Institute of Technology, 2006.
- [22] T. Farr, Five Desktop Publishing Options for Those Looking for an Indesign Alternative, 18.03.2014. Dostupno na: <https://www.udemy.com/blog/indesign-alternative/>. (Pokušaj pristupa 20.09.2014.).
- [23] Adobe Scripting Center, Adobe Systems Incorporated, Dostupno na: <http://www.adobe.com/devnet/scripting.html>. (Pokušaj pristupa 20.09.2014.).
- [24] Adobe Systems Incorporated, Adobe® InDesign® CS6 Scripting Tutorial, Adobe Systems Incorporated, California, USA, 2012.

- [25] Extending QuarkXPress, Quark Software Inc., Dostupno na:
http://www.quark.com/Products/QuarkXPress/Extending_QuarkXPress/. (Pokušaj pristupa 20.09.2014.).
- [26] Programming Publisher 2003 Made Easy: Lesson 1, Microsoft, Dostupno na:
<http://msdn.microsoft.com/en-us/library/aa168501%28v=office.11%29.aspx>. (Pokušaj pristupa 20.09.2014.).
- [27] The Scriptor, Scribus Documentation Project, 10 06 2010. Dostupno na:
http://documentation.scribus.net/index.php/The_Scriptor. (Pokušaj pristupa 20.09.2014.).
- [28] D. Flanagan, JavaScript: The Definitive Guide, Sixth Edition, O'Reilly Media, Inc., 2011.
- [29] Adobe Systems Incorporated, Adobe® InDesign® CS6 Scripting Guide: JavaScript, Adobe Systems Incorporated, California, USA, 2012.
- [30] J. Dongseok, C. Kwang-Moo, Points to Analysis for JavaScript, *SAC '09 Proceedings of the 2009 ACM symposium on Applied Computing, Honolulu*, pp. 1930-1937, 2009.
- [31] Herczeg, Z.; Lóki, G.; Szirbucz, T.; Kiss, Á., Guidelines for JavaScript Programs: Are They Still Necessary?, *Proceedings of the 11th Symposium on Programming Languages and Software Tools (SPLST'09) and 7th Nordic Workshop on Model Driven Software Engineering (NW-MODE'09), Tampere, Finland*, pp. 59-71, 2009.
- [32] J. W. Maxwell, M. MacDonald, T. Nicholson, J. Halpape, S. Taggart, H. Binder, XML Production Workflows? Start with the Web, *EP the journal of electronic publishing*, svez. 13, br. 1, 2010.
- [33] V. Žiljak, K. Pap, J. Žiljak Vujić, J. Lajkovič, Modeling And Simulation As The Basis For Hybridity In The Graphic Discipline Learning/Teaching Area, *Acta Graphica*, br. 20, pp. 31-41, 2009.
- [34] I. Herman, D. Dardailler, SVG Linearization and Accessibility, *Computer Graphics Forum*, svez. 4, br. 21, 2012.
- [35] V. Ceric, New Methods and Tools for the WorldWideWeb Search, *Journal of Computing and Information Technology - CIT* 8, p. 267–276, 2000.
- [36] Y. Yamamoto, K. Kawamura, H. Watanabe, Bit Rate Reduction of Vector Representation of Binary Images, Image Processing, 2006 IEEE International Conference, Atlanta, GA, USA, 2006.

- [37] D. Duce, I. Herman, B. Hopgood, Web 2D Graphics File Formats, *Computer Graphics Forum*, svez. 1, br. 21, 2012.
- [38] J. D. Eisenberg, SVG Essentials, O'Reilly, 2012.
- [39] S. Pavazza, K. Pap, The Alternative Way of Creating Infographics Using SVG Technology, *Acta graphica*, br. 23, pp. 45-56, 2012.
- [40] K. Pap, T. Pavlović, Z. Sabati, M. Barišić, A. Koren, Digital workflow system in graphic production, u *10th International Design Conference Design 2008*, Dubrovnik, Croatia, 2008.
- [41] A. Okanović, Ispitivanje učinkovitosti automatizacije grafičke pripreme primjenom strojnog učenja, diplomski rad, Zagreb: Grafički fakultet u Zagrebu, 2013.
- [42] T. Babnik, Automatizacija Web to Print sustava u digitalnom tisku, diplomski rad, Grafički fakultet Sveučilišta u Zagrebu, 2013.
- [43] E. R. Harold, XML 1.1 Bible, 3rd Edition, Indianapolis, USA: Wiley Publishing, Inc., 2004.
- [44] V. Žiljak, Information system transformation after implementing XML technology, *Proceedings of the 14th International Conference on Information and Intelligent Systems, Varaždin*, pp. 17.-22., 2003.
- [45] G. Gamble, InDesign CS5 Automation Using XML & JavaScript, Leipzig, Germany: TrainingCompany.Com, 2011.
- [46] H. Kipphan, Handbook of Print Media: Technologies and Production Methods, Springer Science & Business Media, 2001.
- [47] K. Jurešić, Reinženjering malog grafičkog poduzeća, diplomski rad, Grafički fakultetu Zagrebu, 2013.
- [48] K. Zickuhr, Rainie L., E--reading rises as device ownership jumps, Pew Research Center, 2014. Dostupno na: http://www.pewinternet.org/files/old-media/Files/Reports/2014/PIP_E-reading_011614.pdf. (Pokušaj pristupa 01. 11. 2014.).
- [49] E. A. Draffan, A. McNaught, A. James, eBooks, Accessibility and the Catalysts for Culture Change, *Computers Helping People with Special Needs, Springer International Publishing*, pp. 543-550, 2014.

- [50] EPUB, International Digital Publishing Forum, Trade and Standards Organization for the Digital Publishing Industry, 2014. Dostupno na: <http://idpf.org/epub>. (Pokušaj pristupa 08. 11. 2014.).
- [51] EPUB and Kindle formats, Dostupno na: <http://digitalpublishing101.com/digital-publishing-101/digital-publishing-basics/epub-kindle-formats/>. (Pokušaj pristupa 01. 11. 2014.).
- [52] J. Ferraiolo, Adventures With SVG In ePub, How To Include Illustrator SVG Files Into eBooks, 10. 2014. Dostupno na: <http://svgmagazine.com/oct2014/Adventures%20With%20SVG%20In%20ePub.html>. (Pokušaj pristupa 06. 11. 2014.).
- [53] J. Schewe, Preparing Images for Delivery - Technical paper, Adobe Systems Incorporated, 2006. Dostupno na: http://www.adobe.com/digitalimag/pdfs/phscs2ip_reproprep.pdf. (Pokušaj pristupa 20. 08. 2014.).
- [54] I. Zjakić, Tehničko uređivanje u procesu izrade knjige, Zagreb: Hrvatska sveučilišna naklada, 2013.
- [55] P. Andrews, Advanced Photoshop Elements 7 for Digital Photographers, Focal Press, Oxford, USA, 2009.
- [56] M. Evening, Adobe Photoshop CS5 for Photographers, Focal Press, Elsevier Ltd, Oxford, UK, 2010.
- [57] M. Autret, Comparing the Performance of ExtendScript Snippets, 2011. Dostupno na: <http://www.indiscripts.com/post/2011/06/comparing-the-performance-of-extendscript-snippets>. (Pokušaj pristupa 19. 09. 2013.).
- [58] M. Kitajima, P. G. Polson, A comprehension-based model of correct performance and errors in skilled, display-based, human-computer interaction, *International Journal of Human-Computer Studies* 43.1, pp. 65-99, 1995.
- [59] P. Kahrel, Scripting InDesign CS3/4 with JavaScript, O'Reilly Media, Inc., 2009.
- [60] S. Souders, Even Faster Websites, Sebastopol, USA: O'Reilly Media Inc., 2009.
- [61] N. C. Zakas, Writing Efficient JavaScript: Chapter 7 - Even Faster Websites, Dostupno na: <http://oreilly.com/server-administration/excerpts/even-faster-websites/writing-efficient-javascript.html>. (Pokušaj pristupa 19. 03. 2013.).

- [62] M. Autret M., On 'everyItem()', 2010. Dostupno na: <http://www.indiscripts.com/post/2010/06/on-everyitem-part-1>. (Pokušaj pristupa 19. 09. 2013.).
- [63] M. Chaney, JPEG images: counting your losses, Dostupno na: <http://www.steves-digicams.com/knowledge-center/jpeg-images-counting-your-losses.html>. (Pokušaj pristupa 02. 05. 2015.).
- [64] B. Johnson, Dostupno na: <http://www.earthboundlight.com/phototips/photoshop-script-listener.html>. (Pokušaj pristupa 02. 05. 2015.).
- [65] C. Bondy, Realizing digital profitability with unified workflow, *International Conference on Digital Printing Technologies*, svez. 2006, pp. 522-532, 2006.
- [66] P. Osmond, The Power of PDF, *Visual Communications Journal*, pp. 29.-31., 2001.

PRILOZI

Tablica 41. Rezultati mjerenja standardnog i skriptnog izvođenja zadatka skripte za brisanje neželjenih bjelina u tekstu

	Podatci o uvezenom dokumentu				Podatci o uređenom dokumentu				Broj testa	Standardan način izvođenja zadatka			Skriptni način izvođenja zadatka		
	Znakova	Riječi	Linija	Odlomaka	Znakova	Riječi	Linija	Odlomaka		Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice	Pritisci miša	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice	Pritisci miša
Knjiga 1	39.398	5.670	630	242	39.348	5.670	493	240	1.	86	11	58	5	0	5
									2.	92	13	62	6	0	5
									3.	70	10	56	5	0	5
									4.	80	0	66	4	0	5
									5.	72	8	60	4	0	5
Knjiga 2	60.914	8.596	1.099	572	60.572	8.596	905	566	1.	78	16	56	6	0	5
									2.	97	8	70	5	0	5
									3.	86	0	72	5	0	5
									4.	76	7	64	7	0	5
									5.	70	4	61	6	0	5
Knjiga 3	399.258	54.335	4.440	1.057	398.360	54.331	4.289	955	1.	86	8	57	14	0	5
									2.	90	8	60	14	0	5
									3.	88	11	58	15	0	5
									4.	94	8	69	14	0	5
									5.	88	9	60	15	0	5
Knjiga 4	789.126	106.001	8.900	2.161	787.466	105.990	8.357	1.993	1.	107	12	56	28	0	5
									2.	109	9	61	27	0	5
									3.	94	9	59	28	0	5
									4.	96	6	61	28	0	5
									5.	104	6	61	28	0	5

Tablica 42. Rezultati mjerenja standardnog i skriptnog izvođenja zadatka - zadržavanje uvezenih znakovnih stilova – 9 kombinacija

	Broj znakova	bold	italic	bold italic	eksponent	indeks	podcrtano	precrtano	small caps	all caps	ukupno	Broj testa	Standardan način izvođenja zadatka				Skriptni način izvođenja zadatka			
													Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice	Pritisci miša	Ukupno	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice	Pritisci miša	Ukupno
Knjiga 1	55.216	41	78	2	15	0	10	0	0	0	146	1.	79	6	40	46	5	0	4	4
												2.	72	6	42	48	5	0	4	4
												3.	76	0	45	45	5	0	4	4
												4.	62	4	41	45	5	0	4	4
												5.	68	4	43	47	5	0	4	4
Knjiga 2	398.360	96	456	1	442	0	171	0	0	0	1.166	1.	81	0	46	46	17	0	4	4
												2.	86	2	45	47	17	0	4	4
												3.	79	5	42	47	17	0	4	4
												4.	71	5	40	45	17	0	4	4
												5.	75	2	44	46	17	0	4	4
Knjiga 3	787.466	131	900	1	747	0	467	0	0	0	2.246	1.	98	0	47	47	35	0	4	4
												2.	95	3	45	48	36	0	4	4
												3.	90	5	43	48	38	0	4	4
												4.	91	7	40	47	36	0	4	4
												5.	99	3	43	46	37	0	4	4

Tablica 43. Rezultati mjerenja skriptnog izvođenja zadatka - zadržavanje uvezenih znakovnih stilova – 144 kombinacije

															Skriptni način izvođenja zadatka			
	Podatci o uvezenom dokumentu Broj znakova	bold + super	italic + underline	italic + super	bold	italic	bold italic	eksponent	indeks	podcrtano	precrtano	small caps	all caps	ukupno	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice	Pritisci miša	Ukupno
Knjiga 1	55.216	0	0	1	41	78	2	15	0	10	0	0	0	147	12	0	4	4
															12	0	4	4
															12	0	4	4
															12	0	4	4
															12	0	4	4
Knjiga 2	398.360	3	1	7	96	456	1	442	0	171	0	0	0	1.177	58	0	4	4
															58	0	4	4
															58	0	4	4
															58	0	4	4
															58	0	4	4
Knjiga 3	787.466	3	2	13	131	900	1	747	0	467	0	0	0	2.264	190	0	4	4
															192	0	4	4
															192	0	4	4
															191	0	4	4
															192	0	4	4

Tablica 44. Rezultati mjerenja standardnog izvođenja zadatka uređivanja tablica u periodičkoj publikaciji

Standardan način izvođenja zadatka															
	Podatci o tablicama		Prelomljena tablica		Broj testa	Utrošeno vrijeme (sekunde)	Vrijeme popravci (sekunde)	Ukupno vrijeme (sekunde)	Pritisci tipki tipkovnice	Pritisci miša	Pritisci tipki tipkovnice popravak	Pritisci miša popravak	Ukupno	Pogreške	
	redaka	stupaca	redaka	stupaca										Broj	opis
Tablica 1	145	9	135	6	1.	668	63	731	367	185	8	20	580	4	Nije postavljen Character Style cijele tablice, nije podešena visina svih redaka, nisu obrisani enteri u tekstu, nisu zadani "header rows"
			135	6	2.	590	16	606	331	174	3	3	511	1	1 x krivo postavljen znakovni stil
			135	6	3.	552	61	613	314	178	10	9	511	3	3 x krivo postavljen znakovni stil
			135	6	4.	699	18	717	328	236	3	6	573	1	1 x krivo postavljen znakovni stil
			135	6	5.	567	199	766	392	162	27	63	644	7	izbrisano 7 redaka tablice
Tablica 2	573	9	531	6	1.	1.196	56	1.252	1.389	332	15	16	1.752	5	5 x krivo postavljen znakovni stil
			531	6	2.	1.215	29	1.244	1.454	324	6	10	1.794	2	2 x krivo postavljen znakovni stil
			531	6	3.	1.126	12	1.138	1.357	320	4	5	1.686	1	1 x krivo postavljen znakovni stil
			531	6	4.	1.135	0	1.135	1.043	412	0	0	1.455	0	nema
			531	6	5.	1.183	19	1.202	1.327	352	4	6	1.689	1	1 x krivo postavljen znakovni stil
Tablica 3	1103	15	1036	6	1.	2.357	113	2.470	2.757	445	18	18	3.238	6	6 x krivo postavljen znakovni stil
			1036	6	2.	2.236	155	2.391	2.659	416	32	28	3.135	10	8 x krivo postavljen znakovni stil, nije postavljen redak u kunama i „header row“
			1036	6	3.	1.990	15	2.005	3.105	341	3	5	3.454	1	1 x krivo postavljen znakovni stil
			1036	6	4.	1.977	18	1.995	1.581	619	3	6	2.209	1	1 x krivo postavljen znakovni stil
			1036	6	5.	2.156	34	2.190	2.151	456	6	9	2.622	2	2 x krivo postavljen znakovni stil

Tablica 45. Rezultati mjerenja skriptnog izvođenja zadatka skripte za uređivanje tablica u periodičkoj publikaciji

	Podatci o tablicama		Prelomljena tablica		Broj testa	Skriptni način izvođenja zadatka						Pogreške	
						redaka	stupaca	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice	Pritisci miša	Ukupno		
												redaka	stupaca
Tablica 1	145	9	135	6	1.	135	6	21	0	4	4	0	nema
			135	6	2.	135	6	20	0	4	4	0	nema
			135	6	3.	135	6	21	0	4	4	0	nema
			135	6	4.	135	6	21	0	4	4	0	nema
			135	6	5.	135	6	21	0	4	4	0	nema
Tablica 2	324	9	305	6	1.	305	6	60	0	4	4	0	nema
			305	6	2.	305	6	58	0	4	4	0	nema
			305	6	3.	305	6	60	0	4	4	0	nema
			305	6	4.	305	6	59	0	4	4	0	nema
			305	6	5.	305	6	59	0	4	4	0	nema
Tablica 3	1103	15	1036	6	1.	531	6	185	0	4	4	0	nema
			1036	6	2.	531	6	184	0	4	4	0	nema
			1036	6	3.	531	6	186	0	4	4	0	nema
			1036	6	4.	531	6	188	0	4	4	0	nema
			1036	6	5.	531	6	186	0	4	4	0	nema

Tablica 46. Rezultati mjerenja skriptnog izvođenja zadatka skripte za izradu numeracije i montaže kartica

Broj polja za numeraciju po kartici	Broj testa	Standardan način izvođenja zadatka		Skriptni način izvođenja zadatka	
		Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša
1	1.	988	24	27	580
	2.	946	27	28	590
	3.	1.005	26	28	563
	4.	968	29	27	552
	5.	992	27	28	556
2	1.	1172	32	53	974
	2.	1153	28	56	994
	3.	1009	35	58	1.023
	4.	1123	33	55	968
	5.	1106	35	55	1.013
3	1.	1391	42	86	1.256
	2.	1311	43	82	1.196
	3.	1293	39	83	1.138
	4.	1309	42	84	1.220
	5.	1331	41	84	1.101
4	1.	1506	55	110	1.386
	2.	1463	53	120	1.379
	3.	1511	49	113	1.326
	4.	1459	47	119	1.356
	5.	1472	55	116	1.402
5	1.	1636	63	158	1.586
	2.	1663	60	155	1.566
	3.	1621	55	152	1.572
	4.	1592	56	159	1.531
	5.	1672	55	148	1.523

Tablica 47. Rezultati mjerenja izvođenja zadatka izrade montaže i personalizacije kartica iz podataka u XML datoteci

Poslovi ubacivanja XML podataka na kartice	Broj testa	Standardan način izvođenja zadatka		Skriptni način izvođenja zadatka	
		Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša
1. Dopisnice Format: 150 x 100 mm Montaža: arak 483 x 330 mm, 9 kartica na arku, 49 araka Napusti: 3 mm XML: 3 XML elementa, 440 kartica	1.	767	347	103	55
	2.	825	371	111	58
	3.	493	282	94	55
	4.	579	299	91	49
	5.	582	260	93	48
2. Akreditacije Format: 63 x 93 mm Montaža: arak 483 x 330 mm, 18 kartica na arku, 5 araka Napusti: 3 mm XML: 4 XML elementa, 83 kartice	1.	556	378	99	53
	2.	491	328	87	58
	3.	683	403	91	51
	4.	508	342	71	51
	5.	475	306	78	56
3. Kodovi Format: 85 x 54 mm Montaža: arak 483 x 330 mm, 25 kartica na arku, 97 araka Napusti: 3 mm XML: 2 XML elementa, 2420 kartica	1.	1224	386	289	41
	2.	856	325	263	43
	3.	1005	344	260	44
	4.	898	282	252	41
	5.	853	261	261	45

Tablica 48. Rezultati mjerenja izvođenja zadatka izrade montaže u programu za prijelom

Poslovi izrade montaže:	Broj testa	Standardan način izvođenja zadatka		Skriptni način izvođenja zadatka	
		Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša
1. Posjetnice Format: 90 x 50 mm Broj stranica: 3 Montaža: arak 483 x 330 mm, 24 na arku (3x8), jednostrano Napusti: 3 mm	1.	355	241	28	15
	2.	285	146	24	12
	3.	252	179	26	13
	4.	214	141	31	17
	5.	207	141	24	13
2. Razglednice Format: 150 x 100 mm Broj stranica: 6 Montaža: arak 483 x 330 mm, 9 na arku (3x3), obostrano Napusti: 3 mm	1.	360	257	26	15
	2.	283	172	23	15
	3.	274	164	26	15
	4.	235	140	27	17
	5.	286	203	29	17
3. Priznanja Format: 210 x 297 mm Broj stranica: 38 Montaža: arak 450 x 320 mm, 2 na arku, jednostrano Napusti: 3 mm	1.	460	385	49	15
	2.	602	436	44	11
	3.	389	326	50	12
	4.	392	319	44	11
	5.	418	371	42	11

Tablica 49. Rezultati mjerenja izvođenja zadatka izrade ePub dokumenta koji sadrži SVG datoteke

Poslovi izrade ePub dokumenta	Broj testa	Standardan način izvođenja zadatka		Skriptni način izvođenja zadatka	
		Utrošeno vrijeme (sekunde)	Pritisци tipki tipkovnice i miša	Utrošeno vrijeme (sekunde)	Pritisци tipki tipkovnice i miša
1. Knjiga Broj poveznica za zamjenu SVG datotekama: 10.ai datoteka	1.	980	598	120	57
	2.	701	510	131	61
	3.	690	498	123	56
	4.	820	550	120	62
	5.	756	528	133	52
2. Knjiga Broj poveznica za zamjenu SVG datotekama: 16.eps datoteka	1.	1333	942	135	52
	2.	1298	837	140	60
	3.	1168	871	134	65
	4.	1256	887	118	54
	5.	1213	841	124	56

Tablica 50. Rezultati mjerenja izvođenja zadatka obrade fotografija

Poslovi obrade fotografija u dokumentu	Broj testa	Standardan način izvođenja zadatka		Skriptni način izvođenja zadatka	
		Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša	Utrošeno vrijeme (sekunde)	Pritisci tipki tipkovnice i miša
1. Brošura Br. poveznica: 17 Br. JPEG slika: 13 RGB Slike s eppi < 300: 6 JPEG Poveznice koje se ne obrađuju: 2 PDF, 2 PSD	1.	521	187	58	5
	2.	547	238	55	5
	3.	562	287	55	5
	4.	538	229	56	5
	5.	552	218	56	5
2. Časopis Br. poveznica: 108 Br. JPEG slika: 64 RGB + 1 greyscale + 10 CMYK Slike s eppi < 300: 31 JPEG Poveznice koje se ne obrađuju: 2 AI, 20 PDF, 11 PSD	1.	2551	1.058	292	5
	2.	2295	1.032	299	5
	3.	2321	1.091	296	5
	4.	2496	1.117	299	5
	5.	2383	1.083	291	5

ŽIVOTOPIS

Mr. sc. Sanja Brekalo viši je predavač na Međimurskom veleučilištu u Čakovcu. Rođena je 1979. godine u Čakovcu. Studirala je na Grafičkom fakultetu u Zagrebu gdje je diplomirala i magistrirala. Nakon završetka studija više godina radi poslove struke u Tiskari Zrinski d.d. te kasnije u vlastitom poduzeću Graphprint d.o.o. Predavala je u Graditeljskoj školi u Čakovcu, Elektrostrojarskoj školi u Varaždinu, na Institutu cjeloživotnog učenja i novih obrazovnih tehnologija u Varaždinu te na Veleučilištu u Varaždinu.

Na Međimurskom veleučilištu u Čakovcu izabrana je u zvanje višeg predavača 2014. godine. Tijekom svog rada bavi se istraživanjem, razvojem i primjenom računala i grafičkih tehnologija u područjima izrade web stranica, obrazovanja pomoću e-učenja, grafičke pripreme, tiska i dorade.

Objavljeni radovi:

1. Brekalo, Sanja; Pap, Klaudio, Stanić, Nikolina.
Optimisation of automatic variable graphic layout and imposition // Tehnički vjesnik/Technical Gazette. StrojarSKI fakultet, Slavonski Brod, prihvaćen za objavu u Vol. 23/No. 1, 2016. (Print: ISSN 1330-3651, Online: ISSN 1848-6339)
2. Brekalo, Sanja; Pap, Klaudio.
Optimizacija procesa izrade digitalne grafičke pripreme za varijabilni digitalni tisak barkodova korištenjem skriptiranja // Polytechnic & Design, Tehničko veleučilište u Zagrebu, Vol. 3 (2015), No. 1, p. 109-118 ISSN: 1849-1195
3. Breslauer, Nenad; Brekalo, Sanja; Breslauer, Nevenka
Eduaroam Grad Čakovec i MEV // Međunarodni znanstveni skup Tiskarstvo & dizajn 2014., p. 192-197, ISBN 978-953-7064-23-5
4. Brekalo, Sanja; Knok, Željko; Breslauer, Nenad.
Optimizacija procesa prijeloma publikacija korištenjem skriptiranja // Međunarodni znanstveni skup Tiskarstvo & dizajn 2013., p. 105-111, ISBN 978-953-7064-20-4
5. Brekalo, Sanja; Horvatić, Sebastijan.
Usporedba značajki web animacija kreiranih u SVG, HTML5 Canvas i Flash tehnologijama // Zbornik radova Međimurskog veleučilišta u Čakovcu, p. 29-37, Vol.4 No.2 Prosinac 2013., ISSN 1847-702X (Tisak), ISSN 1849-1138 (Online)

6. Brekalo, Sanja; Čemerika, Denis.
Usporedba video formata // Zbornik radova Međimurskog veleučilišta u Čakovcu, p. 23-27, Vol.4 No.2 Prosinac 2013., ISSN 1847-702X (Tisak), ISSN 1849-1138 (Online)
7. Brekalo, Sanja; Breslauer, Nevenka; Breslauer, Nenad.
Implementacija moodle-a na Međimurskom veleučilištu u Čakovcu // Međunarodni znanstveni skup Tiskarstvo & Design 2012., p. 136 ISBN 978-953-7064-18-1
8. Brekalo, Sanja; Breslauer, Nevenka; Breslauer, Nenad.
Korištenje Moodle-a pri izradi elektroničkih obrazovnih sadržaja te održavanju nastave na daljinu // Međunarodni znanstveni skup Tiskarstvo & Design 2012., p. 137 ISBN 978-953-7064-18-1
9. Brekalo, Sanja; Vrtarić, Matija.
Optimizacija web stranica za internet pretraživače // Zbornik radova Međimurskog veleučilišta u Čakovcu, p. 21-25, Vol.2 No.2 Prosinac 2011., ISSN 1847-702X (Tisak), ISSN 1849-1138 (Online)
10. Brekalo, Sanja; Knok, Željko.
Digital technology application in production of labels and stickers in rolls // Tiskarstvo 2010 (ed. V. Žiljak), Stubičke toplice-Zagreb: Akademija tehničkih znanosti hrvatske, Centar za grafičko inženjerstvo, Grafički fakultet, 2010., ISBN 978-953-7064-14-3
11. Brekalo, Sanja.
Finishing effects in digital printing – clear toner // Tiskarstvo 09 (ed. V. Žiljak), Stubičke toplice - Zagreb: Akademija tehničkih znanosti hrvatske, Centar za grafičko inženjerstvo, Grafički fakultet, 2009. ISBN: 978-953-7064-12-2
12. Brekalo, Sanja; Babić, Darko.
Technological dependance of folding on the sort and quality of paper in finishing processes - folding 12 pages folds // Tiskarstvo 09 (ed. V. Žiljak), Stubičke toplice - Zagreb: Akademija tehničkih znanosti hrvatske, Centar za grafičko inženjerstvo, Grafički fakultet , 2009. ISBN: 978-953-7064-12-2
13. Brekalo, Sanja; Babić, Darko.
Technological dependance of folding on the sort and quality of paper in finishing processes - folding 16 pages folds // Zbornik radova 12. th International conference on printing, design and graphic communications “Blaž Baromić“. 2008. p. 59-63 ISBN: 987-953-96020-9-1

14. Brekalo, Sanja.

Utjecaj tiskovne forme na kvalitetu otiska // Zbornik radova 8. Savjetovanja tiskarstva, dizajna i grafičkih komunikacija "Blaž Baromić". 2004. p. 207-212, ISBN 953-96020-2-5