

2020 < ŠTEVILKA 4 < OKT. NOV. DEC. < LETNIK XXVIII < ISSN 1318-1882

04

U P O R A B N A

INFORMATIKA

U P O R A B N A I N F O R M A T I K A

2020 ŠTEVILKA 4 OKT/NOV/DEC LETNIK XXVIII ISSN 1318-1882

▣ Strokovni prispevki

- Bor Krizmanič, Aleš Groznik
Aktualne dejavnosti, primerne za uporabo veriženja podatkovnih blokov 175
- Mitja Gradišnik, Tina Beranič, Sašo Karakatič
Z umetno inteligenco podprt razvoj programske opreme 186
- Mateja Kocbek Bule
Znanjski delavci in notacija CMMN 196

▣ Kratki znanstveni prispevki

- Amra Omanović, Polona Oblak, Tomaž Curk
Application of tropical semiring for matrix factorization 205
- Matej Vitek, Peter Rot, Vitomir Štruc, Peter Peer
Poglobljen pogled v beločnično biometrijo: nova podatkovna množica in študija delovanj 209
- Vida Groznik
Digitalizacija in prilagoditev psihološkega testa za uporabo s sistemom za spremljanje očesnih gibov 215

▣ Informacije

- Iz slovarja** 224

Ustanovitelj in izdajatelj

Slovensko društvo INFORMATIKA
Litostrojska cesta 54, 1000 Ljubljana

Predstavniki

Niko Schlamberger

Odgovorni urednik

Saša Divjak

Uredniški odbor

Andrej Kovačič, Evelin Krnac, Ivan Rozman, Jan Mendling, Jan von Knop, John Taylor, Jurij Jaklič, Lili Nemeč Zlatolas, Marko Hölbl, Mirjana Kljajić Borštnar, Mirko Vintar, Pedro Simões Coelho, Saša Divjak, Sjaak Brinkkemper, Slavko Žitnik, Tatjana Welzer Družovec, Vesna Bosilj-Vukšić, Vida Groznik, Vladislav Rajkovič

Recenzentski odbor

Alenka Kavčič, Andrej Brodnik, Andrej Kovačič, Bor Plestenjak, Borut Werber, Borut Žalik, Boštjan Žvanut, Božidar Potočnik, Ciril Bohak, Danijel Skočaj, David Jelenc, Dejan Georgiev, Dejan Lavbič, Denis Trček, Dobravec Tomaž, Domen Mongus, Eva Krhač, Franc Solina, Gregor Weiss, Igor Kononenko, Janez Demšar, Jurij Jaklič, Jurij Mihelič, Katarina Puc, Lovro Šubelj, Luka Čehovin, Luka Pavlič, Marko Bajec, Marko Hölbl, Marjan Heričko, Marjan Krisper, Martin Vodopivec, Matevž Pesek, Matija Marolt, Mihaela Triglav Čekada, Mirjana Kljajić Borštnar, Mojca Indihar Štemberger, Monika Klun, Peter Trkman, Sandi Gec, Saša Divjak, Slavko Žitnik, Tomaž Erjavec, Uroš Godnov, Uroš Rajkovič, Vida Groznik, Vladislav Rajkovič, Vlado Stankovski, Živa Rant

Tehnični urednik

Slavko Žitnik

Lektoriranje angleških izvlečkov

Marvelingua (angl.)

Oblikovanje

KOFEIN DIZAJN, d. o. o.

Prelom in tisk

Boex DTP, d. o. o., Ljubljana

Naklada

200 izvodov

Naslov uredništva

Slovensko društvo INFORMATIKA
Uredništvo revije Uporabna informatika
Litostrojska cesta 54, 1000 Ljubljana
www.uporabna-informatika.si

Revija izhaja četrtletno. Cena posamezne številke je 20,00 EUR. Letna naročnina za podjetja 85,00 EUR, za vsak nadaljnji izvod 60,00 EUR, za posameznike 35,00 EUR, za študente in seniorje 15,00 EUR. V ceno je vključen DDV.

Revija Uporabna informatika je od številke 4/VII vključena v mednarodno bazo INSPEC.

Revija Uporabna informatika je pod zaporedno številko 666 vpisana v razvid medijev, ki ga vodi Ministrstvo za kulturo RS.

Revija Uporabna informatika je vključena v Digitalno knjižnico Slovenije (dLib.si).

© Slovensko društvo INFORMATIKA

Vabilo avtorjem

V reviji Uporabna informatika objavljamo kakovostne izvirne članke domačih in tujih avtorjev z najširšega področja informatike v poslovanju podjetij, javni upravi in zasebnem življenju na znanstveni, strokovni in informativni ravni; še posebno spodbujamo objavo interdisciplinarnih člankov. Zato vabimo avtorje, da prispevke, ki ustrezajo omenjenim usmeritvam, pošljejo uredništvu revije po elektronski pošti na naslov ui@drustvo-informatika.si.

Avtorje prosimo, da pri pripravi prispevka upoštevajo navodila, objavljena v nadaljevanju ter na naslovu <http://www.uporabna-informatika.si>.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Članki so anonimno recenzirani, o objavi pa na podlagi recenzij samostojno odloča uredniški odbor. Recenzenti lahko zahtevajo, da avtorji besedilo spremenijo v skladu s priporočili in da popravljeni članek ponovno prejmejo v pregled. Uredništvo pa lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če članek ne ustreza kriterijem za objavo v reviji.

Pred objavo članka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost članka in dovoljuje prenos materialnih avtorskih pravic. Nenaročenih prispevkov ne vračamo in ne honoriramo. Avtorji prejmejo enoletno naročnino na revijo Uporabna informatika, ki vključuje avtorski izvod revije in še nadaljnje tri zaporedne številke.

S svojim prispevkom v reviji Uporabna informatika boste prispevali k širjenju znanja na področju informatike. Želimo si čim več prispevkov z raznoliko in zanimivo tematiko in se jih že vnaprej veselimo.

Uredništvo revije

Navodila avtorjem člankov

Članke objavljamo praviloma v slovenščini, članke tujih avtorjev pa v angleščini. Besedilo naj bo jezikovno skrbno pripravljeno. Priporočamo zmernost pri uporabi tujk in – kjer je mogoče – njihovo zamenjavo s slovenskimi izrazi. V pomoč pri iskanju slovenskih ustreznih priporočamo uporabo spletnega terminološkega slovarja Slovenskega društva Informatika Islovar (www.islovar.org).

Znanstveni članek naj obsega največ 40.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Članek naj bo praviloma predložen v urejevalniku besedil Word (*.doc ali *.docx) v enojnem razmaku, brez posebnih znakov ali poudarjenih črk. Za ločilom na koncu stavka napravite samo en prazen prostor, pri odstavkih ne uporabljajte zamika.

Naslovu članka naj sledi za vsakega avtorja polno ime, ustanova, v kateri je zaposlen, naslov in elektronski naslov. Sledi naj povzetek v slovenščini v obsegu 8 do 10 vrstic in seznam od 5 do 8 ključnih besed, ki najbolje opredeljujejo vsebinski okvir članka. Pred povzetkom v angleščini naj bo še angleški prevod naslova, prav tako pa naj bodo dodane ključne besede v angleščini. Obratno velja v primeru predložitve članka v angleščini. Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Vsako sliko in tabelo razložite tudi v besedilu članka. Če v članku uporabljate slike ali tabele drugih avtorjev, navedite vir pod sliko oz. tabelo. Revijo tiskamo v črno-beli tehniki, zato barvne slike ali fotografije kot original niso primerne. Slik zaslonov ne objavljamo, razen če so nujno potrebne za razumevanje besedila. Slike, grafikoni, organizacijske sheme ipd. naj imajo belo podlago. Enačbe oštevilčite v oklepajih desno od enačbe.

V besedilu se sklicujte na navedeno literaturo skladno s pravili sistema APA navajanja bibliografskih referenc, najpogosteje torej v obliki (Novak & Kovač, 2008, str. 235). Na koncu članka navedite samo v članku uporabljeno literaturo in vire v enotnem seznamu po abecednem redu avtorjev, prav tako v skladu s pravili APA. Več o sistemu APA, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani <http://owl.english.purdue.edu/owl/resource/560/01/>.

Članku dodajte kratek življenjepis vsakega avtorja v obsegu do 8 vrstic, v katerem poudarite predvsem strokovne dosežke.

█ Aktualne dejavnosti, primerne za uporabo veriženja podatkovnih blokov

Bor Krizmanič¹, Aleš Groznik¹

¹Univerza v Ljubljani, Ekonomska fakulteta, Kardeljeva ploščad 17, 1000 Ljubljana, Slovenija
bor.krizmanic@ef.uni-lj.si, ales.groznik@ef.uni-lj.si

Izvleček

Tehnologija veriženja podatkovnih blokov (angl. Blockchain) se v zadnjih letih omenja v povezavi z uporabo v najrazličnejših panogah in za zelo raznolike načine uporabe. Težko je presoditi, katera področja so resnično najbolj primerna za uporabo te tehnologije. V članku smo obdelali osnovne lastnosti, prednosti in slabosti tehnologije veriženja podatkovnih blokov in predstavili štiri, po naši oceni najprimernejša področja dejavnosti za uporabo te tehnologije. Potencial za uporabo te tehnologije smo po posameznih področjih ocenjevali na podlagi pregleda pravnega, tehnološkega in sociološkega vidika morebitne implementacije ter idej in primerov uporabe. Po našem mnenju so za uporabo najperspektivnejša področja: finančne in zavarovalniške dejavnosti, dejavnost prometa in skladiščena, trgovina, vzdrževanje in popravila motornih vozil ter kot zadnje, informacijske in komunikacijske dejavnosti.

Ključne besede: dokaz o deležu, dokaz o delu, kriptovalute, tehnologija razpršene evidence, tehnologija veriženja podatkovnih blokov

Abstract

In recent years, the use of a blockchain technology has been discussed in connection with a variety of industries and different applications. It is difficult to judge which areas are best suited for the use of this technology. In this article we have described the basic characteristics, advantages and disadvantages of blockchain technology and presented four economic sectors which we believe are most suitable for the use of this technology. For each sector, we evaluated the potential for the use of this technology based on a review of the legal, technological and sociological aspects of possible implementation, ideas and applications. In our view, the most promising areas of application are financial and insurance activities; transporting and storage; wholesale and retail; repair of motor vehicles and motorcycles; information and communication services.

Keywords: proof of work, proof of stake, cryptocurrency, distributed ledger technology, blockchain.

1 UVOD

Tehnologija veriženja podatkovnih blokov je vse od njenega pojava v letu 2008, s prvo zasnovano platforme Bitcoin za namen delovanja istoimenske kriptovalute, deležna velike pozornosti tako laične, kot tudi strokovne javnosti. Čeprav je pozornost v zadnjem letu, po občutnem padcu vrednoti večine kriptovalut generalno upadla, pa se še vedno namenja veliko pozornosti uporabi tehnologije veriženja podatkovnih blokov v namene, ki niso povezani s kriptovalutami. Cilj tega članka je definirati tiste dejavnosti, ki so najperspektivnejše za uporabo te tehnologije. Ker je iz objavljene strokovne literature razvidno, da se predvideva uporaba te tehnologije v skoraj vseh dejavno-

stih, je zato namen tega članka trezno in preudarno preučiti, katere dejavnosti so primerne za uporabo in katere ne. V luči tega je v tem članku podan krajši pregled najprimernejših dejavnosti za uporabo te tehnologije, opredeli pa se tudi tistih, ki morda niso najbolj primerne, a so vendarle deležne velikega zanimanja javnosti.

S tehnologijo veriženja podatkovnih blokov so se do sedaj ukvarjali in jo tudi sooblikovali številni pomembni avtorji s tega področja. Tako je to tehnologijo prvi zasnoval in opisal skrivnostni snovalec Bitcoina pod psevdonimom Satoshi Nakamoto. Pred njim se je s podobnimi koncepti ukvarjal in o njih pisal ameriški pravnik in računalniški znanstvenik Nick

Szabo, ki je tudi zasnoval, v več pogledih predhodnika Bitcoin, digitalno valuto Bit Gold¹. Po uveljavitvi Bitcoin in z začetki širšega zanimanja za tehnologijo, ki ga omogoča, sta postala vplivna avtorja še Alex in Don Tapscott.

2 RAZLAGA POJMOV

Tehnologija veriženja podatkovnih blokov (angl. Blockchain) lahko opišemo kot tehnični protokol, ki omogoča izmenjavo podatkov in vzpostavitev zaupanja brez potrebe po centralnih posrednikih (Seffinga, Lyons & Bachmann, 2017, str. 7). Informacije se shranjujejo v verigo podatkovnih blokov, ki vsebujejo niz transakcij oz. podatkov, ki so nastali v času od nastanka predhodnega bloka. V osnovni zasnovi tehnologije veriženja podatkovnih blokov se za izdelavo blokov uporablja t. i. protokol dokaza o delu (angl. proof-of-work), pri katerem vozlišča (tj. udeleženci omrežja), imenovana rudarji, uporabljajo t. i. zgoščevalno funkcijo (angl. hash function) z namenom iskanja pravilne zgoščene vernosti (angl. hash). Zgoščevalna funkcija šifrira vhodne podatke poljubne dolžine v standardno število znakov (številk in črk). Ko rudarji najdejo zgoščeno vrednost, ki ustreza zahtevam, izdelajo nov blok. Vsak blok vsebuje to zgoščeno vrednost vseh transakcij oz. podatkov, ki so nastali v določenem obdobju, spremenljivko imenovano nonce in zgoščeno vrednost predhodnega bloka. Vsak nov blok se javno objavi, nato pa ostala vozlišča pregledajo ali je bil blok izdelan pravilno in ga potrdijo s konsenzom večine vozlišč. Vsak novi blok se sklicuje na zgoščeno vrednost predhodnega bloka in tako tvori verigo. Vsak zapis je preko podatkovnih blokov dodan nespremenljivi verigi in deljen vsem vozliščem omrežja (Nakamoto, 2008, str. 1–3). Če bi želel nekdo spremeniti katerikoli zapis, ki je bil že zgoščen v blok v verigi, bi moral na novo izračunati zgoščeno vrednost tega in vseh drugih blokov v verigi. To bi bilo računsko izjemno zahtevno in v večjem omrežju z velikim številom zapisov domala nemogoče. Protokol omogoča dostop in preverjanje vseh zapisov, vse od prvega zapisa na verigi dalje. Ta tehnologija tako omogoča nespremenljiv, deljen, pregleden in preverljiv zapis vseh zapisov (Reyna, Martín, Chen, Soler & Díaz, 2018, str. 174).

Kot alternativo dosedanemu protokolu dokaza o delu poizkušajo številni projekti razviti praktično uporaben in delujoč protokol dokaza o deležu (angl. proof-of-stake), pri katerem ne izdelujejo novih blokov vsa vozlišča v omrežju naenkrat, temveč se iz nabora potrjevalcev po izbranem ključu izmenjujejo vozlišča, ki izdelajo in potrdijo nov blok. S tem se zmanjša računsko in energetska zahtevnost izdelave novega bloka (Ethereum Foundation, 2020). Pogostejša omenjena alternativa dokaza o delu je tudi protokol dokaza o avtoriteti (angl. proof-of-authority), pri katerem nove bloke potrjujejo le v naprej preverjeni in potrjeni uporabniki (imenovani tudi potrjevalci). Ker se sistem zanaša na ugled uporabnika, se mu lahko ob neprimernem delovanju tudi odvzame pravice za potrjevanje. Dokazu o deležu, z nekaj podobnostmi dokaza o avtoriteti, je precej podoben tudi delegiranemu dokazu o deležu (angl. delegated proof-of-stake), kjer se glede na lastništvo kovancev razporedijo glasovalne pravice potrjenih uporabniku o tem kdo bo izdelal nov blok (Ramuta, 2018). Dokaz o kapaciteti (angl. proof-of-capacity) ali tudi dokaz o prostoru (angl. proof-of-space) je zelo podoben konceptu dokaza o delu, le da mora v tem primeru vozlišče namesto računsko moč vložiti veliko prostora na trdem disku. Ta koncept naj bi bil predvsem manj energetsko potrošen in zato bolj prijazen okolju kot dokaz o delu (Hayes, 2020). Sledi še protokol, ki bi ga lahko prevedli kot dokaz o izgorevanju ali kurjenju (angl. proof-of-burn) kjer morajo vozlišča za potrjevanje žrtvovati kovance, proporcionalno s količino porabljenih kovancev pa pridobijo pravice za izdelavo novega bloka. Kurjenje se tehnično izvede tako, da se kovance nakaže na poseben naslov, ki kovance le sprejema, ne pa tudi vrača. Tako postanejo pokurjeni kovanci za vedno nedostopni (Karantias, Kiayias & Zindros, 2020, str. 523–524).

Omrežja veriženja podatkovnih blokov se delijo tudi po tem, kakšne so omejitve za dostop do omrežja. Najbolj pogost in sprva tudi edini tip omrežja je javna veriga podatkovnih blokov (angl. public blockchain), ki dopušča, da se omrežju lahko pridruži kdorkoli iz svetovnega spleta, brez potrebne predhodne prošnje in odobritve za pristop k omrežju, ter brez kakršnega koli preverjanja pristnosti identitete. Najpomembnejši primer takšnega tipa omrežja je seveda Bitcoin. Ker se lahko omrežju pridruži kdorkoli in ga lahko tudi kadarkoli zapusti, so tovrstna omrežja načeloma zasnovana tako, da ne računajo na

¹ <http://unenumerated.blogspot.com/2005/12/bit-gold.html> (zadnji obisk 25.8.2020)

samovoljno pripravljenost udeležencev za vzdrževanje omrežja, temveč udeležbo pri tem nagrajuje s spodbudami za ohranjanje omrežja. Tako na primer v primeru protokola dokaza o delu, sistem nagrajuje rudarje za njihov doprinos k delovanju omrežja (Lai & Lee Kuo Chuen, 2018, str. 153).

Obstaja pa tudi tip zasebne verige podatkovnih blokov (angl. private blockchain), ki za priključitev omrežju zahteva predhodno preverjanje pristnosti identitete in odobritev dostopa (Lai & Lee Kuo Chuen, 2018, str. 153–154). Tovrstna omrežja predpostavljajo, da so vsi udeleženci znani ter imajo pravico in motivacijo za udeležbo, zato ni nujno potrebno ustvarjati notranje spodbude kot pri javni blokovni verigi (Lai & Lee Kuo Chuen, 2018, str. 154). Mehanizmi nadzora dostopa se razlikujejo. Tako lahko o novih udeležencih odločajo obstoječi udeleženci ali pa nekakšen regulatorni organ, neke vrste administrator omrežja, ki izdaja dovoljenja za sodelovanje (Jayachandran, 2017). Poleg osnovne delitve na javne in zasebne verige podatkovnih blokov obstajajo tako še druge, vmesne delitve. Med njimi se največkrat omenja t. i. konzorcij, kjer je protokol konsenza nadzorovan s strani vnaprej izbranih, preverjenih vozlišč, ki potrjujejo nove bloke. Kdo lahko bere te bloke, se lahko omejuje, ali pa tudi ne. Ena od vmesnih rešitev je tudi delno zasebna veriga podatkovnih blokov, kjer eno podjetje ali organizacija dodeljuje dostop uporabnikom, ki izpolnjujejo pogoje zanj. Vse te različne oblike omejenega dostopa se načeloma uporabljajo pri uporabi tehnologije veriženja podatkovnih blokov med podjetji (Dobson, 2018).

Pomemben koncept v tehnologiji veriženja podatkovnih blokov so tudi t. i. pametne pogodbe (angl. smart contract). V pravnem smislu ne gre nujno za pravo pogodbo temveč gre za računalniško kodo, ki je vgrajena v verigo podatkovnih blokov in je posledično na enak način, kot to velja za samo verigo, tudi sama nespremenljiva (Reed, Sathyanarayan, Ruan & Collins, 2018, str. 161). S tem se lahko brez udeležbe zaupanja vredne tretje stranke omogoča izvrševanje sporazumov med strankami, ki si med seboj ne zaupajo (Alharby & Moorsel, 2017). Za integracijo pametnih pogodb v tehnologijo veriženja podatkovnih blokov so se oblikovale številne nove platforme, med katerimi je najpomembnejša platforma Ethereum², ki za pisanje pametnih pogodb uporablja za

to razvit programski jezik Solidity (Lai & Lee Kuo Chuen, 2018, str. 172-173). Za samo izvedbo pogodbenih klavzul je potreben nek sprožilec oz. signal, zato pametne pogodbe potrebujejo verodostojen dostop do podatkov o stanjih in dogodkih v realnem svetu. Te vire podatkov imenujejo t. i. preroki (angl. oracles), ki so ključni za uspešno integracijo pametnih pogodb v realnem svetu. Tako ima na primer, platforma Ethereum že nekaj varnih virov, ki črpajo podatke iz zaupanja vrednih spletnih mest, vendar ti viri jamčijo točnost podatkov zgolj na podlagi ugle-da upravljavcev teh spletnih mest (Zhang, Cecchetti, Croman, Juels & Shi, 2016, str. 270).

Decentralizirane aplikacije (angl. decentralized application) ali pogosto imenovane DApps so na aplikacije, ki tečejo na decentraliziranem omrežju tipa »vsak z vsakim«. Omrežja tehnologija veriženja podatkovnih blokov, kot je na primer Ethereum, predstavljajo zelo primeren porazdeljen sistem za njihovo delovanje. To pomeni, da takšne aplikacije nimajo kakšnega centralnega strežnika, kar bi lahko ob nedelovanju povzročilo nedelovanje celotne aplikacije (Bambara, Allen, 2018, str. 233-234).

3 PREDNOSTI IN SLABOSTI TEHNOLOGIJE VERIŽENJA PODATKOVNIH BLOKOV

Glavna prednost tehnologije veriženja podatkovnih blokov je zmožnost vzpostavitve zaupanja v decentraliziranem okolju, brez centralnih posrednikov. Njen protokol rešuje t. i. problem dvojne porabe znotraj decentraliziranega omrežja tipa »vsak z vsakim« (angl. peer-to-peer). Tako ta tehnologija omogoča vzpostavitev konsenza o eni zgodovini dogodkov in sicer v omrežju z udeleženci, ki se med seboj ne poznajo in si ne zaupajo, in to brez pomoči tretje, zaupanja vredne stranke (Nakamoto, 2008, str. 1–2).

Med prednosti te tehnologije lahko štejemo tudi sledljivost in preverljivost. Vse transakcije ali podatki se namreč javno objavijo in za vedno zapišejo v vseh glavnih knjigah udeležencev oz. tistih udeležencev, ki se odločijo hraniti celotno knjigo. Vse to omogoča, da so vsi zapisi oz. celotna zgodovina v verigi podatkovnih blokov kadarkoli vidni vsem članom omrežja. To poleg sledljivosti in preverljivosti pomeni tudi, da tehnologija veriženja podatkovnih blokov ne omogoča brisanja transakciji ali podatkov – to zadnje pa predstavlja istočasno tudi oviro za določene primere uporabe (Hooper, 2018). To ima v prvi vrsti vpliv na uporabo povsod, kjer so vklju-

² <https://ethereum.org/> (zadnji obisk 20.8.2020)

čeni osebni podatki- več o tem v odstavku glede zakonskih omejitev. Podobno pa velja tudi za lastnost transparentnosti, ki se prav tako pogosto omenja kot prednost. Tako je na primer v omrežju Bitcoin moč videti vse transakcije in ključne podatke o njih (npr. količina prenesenih kovancev, naslove spletnih denarnic) za celotno zgodovino omrežja. To ima tako pozitivne učinke na preglednost omrežja kot tudi ovira, za na primer ohranjanja poslovnih skrivnosti (Reyna, Martín, Chen, Soler & Díaz, 2018, str. 176).

Na drugi strani pa se tehnologija veriženja podatkovnih blokov sooča s številnimi ovirami oz. slabostmi. Kot prvo gre izpostaviti problem omejene razširljivosti oz. omejene kapacitete omrežja. Veriga podatkovnih blokov z novimi zapisi le raste in se nikoli ne krajša, saj so, kot že omenjeno, vsi zapisi za vedno vdelani v verigo podatkovnih blokov. Zaradi tega morajo vozlišča v omrežju shranjevati vedno večjo glavno knjigo, kar zahteva vedno več strojnih in mrežnih virov (Reyna, Martín, Chen, Soler & Díaz, 2018, str. 177). V času največjega zanimanja za kriptovalute tekom leta 2018 je Bitcoinova veriga podatkovnih blokov rastla s hitrostjo okoli 50 GB na leto (Bank for International Settlements, 2018, str. 99). Novembra 2019 pa je veriga blokov Bitcoin preseгла velikost kar 230 GB (Bitcoin.com, 2019). Za primerjavo - uveljavljena plačilna sistema Mastercard in Visa opravita oba skupaj med 3500 in 2000 transakciji na sekundo (angl. transactions per second- TPS), v enakem času pa omrežje Bitcoin izvede skoraj tisočkrat manj transakcij (Bank for International Settlements, 2018, str. 99). Praktično vsa omrežja verige podatkovnih blokov na tem področju iščejo izboljšave, v skladu s tem pa se je pojavilo nekaj potencialnih rešitev. Že samo omrežje Bitcoin je zasnovano tako, da celotno glavno knjigo hranijo le t. i. polna vozlišča (angl. full nodes), ki lahko v celoti potrdijo transakcijo. Kapaciteto omrežja se lahko prilagaja tudi s spreminjanjem zahtevnosti izdelave novega bloka in velikostjo nagrade za rudarje, ki vpliva na interes le-teh za izdelavo novih blokov (ConsenSys, 2019). Kot rešitev se predlaga tudi protokol konsenza na principu dokaza o deležu. Prav slednji protokol bo uporabljala tudi nova verzija omrežja Ethereum (Ethereum 2.0) in s tem po napovedih zelo močno povečala kapaciteto omrežja (Ethereum, 2020). Omenjajo se še t. i. rešitve izvajale transakcije izven verige (angl. off-chain transactions), kar bi povečalo kapaciteto na račun večje verjetnosti izgube podat-

kov. Ena izmed predlaganih rešitev je tudi zmanjšanje časovnega zamika pri potrjevanju novih blokov, kar bi skrajšalo čas potovanja transakcij na račun zmanjšane varnosti (Reyna, Martín, Chen, Soler & Díaz, 2018, str. 174–175).

Drugo veliko oviro pri širši uporabi tehnologije veriženja podatkovnih blokov v različnih dejavnostih pa predstavljajo zakonske omejitve in negotovosti, pri katerih prevladujejo omejitve v povezavi z varstvom osebnih podatkov. Zaradi leta 2016 sprejete Uredbe Evropskega parlamenta in Sveta o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov (Ur. l. EU št. 2016/679) je možnost uporabe te tehnologije za hranjenje osebnih podatkov močno vprašljiva. Ker je omrežje tehnologije veriženja podatkovnih blokov decentralizirano, ni mogoče določiti pooblaščenih oseb za varstvo osebnih podatkov in obdelovalca podatkov, kot to zahteva omenjena uredba v primeru hranjenja osebnih podatkov. Težko bo tudi doseči omejitev prenašanja osebnih podatkov izven Evropske unije, ki z izjemo nekaj določenih držav ni dovoljena. Prav tako velik problem predstavlja načelo, da je vsak posameznik lastnik svojih osebnih podatkov in lahko kadarkoli zahteva izbris le-teh. Glede na to, da tehnologija veriženja podatkovnih blokov vse podatke deli med vsa vozlišča v omrežju, ki lahko te podatke shranijo, hkrati pa ne omogoča brisanja češarkoli že shranjenega v verigi podatkovnih blokov, se zdi, da osebnih podatkov ne bo moč shranjevati na verigi podatkovnih blokov (Millard, 2018, str. 845).

Z zakonodajo pa je omenjena tudi uporaba tehnologije veriženja podatkovnih blokov za beleženje lastninskih in drugih pravic na sredstvih, ki niso del verige podatkovnih blokov (kot so kriptovalute) in se nahajajo izven le-te. V osnovni zasnovi te tehnologije namreč ni možnosti identifikacije strank v transakcijah (v osnovni zasnovi so anonimizirane) in ni mogoče spreminjati zapisov o pravicah na sredstvih s strani tretje, zaupanja vredne stranke, kot bi bilo to potrebno na primer ob sodni odločbi o zaplembi premoženja. Sedanja zasnova za vsako transakcijo namreč zahteva soglasje takratnega lastnika (Reed, Sathyanarayan, Ruan & Collins, 2018, str. 170–181). Ker je ta tehnologija dokaj nova in precej drugačna od katerekoli druge, se sodna praksa v povezavi z njo šele oblikuje. To povzroča negotovost glede možnosti njene uporabe (Bacon, Michels, Millard, & Singh, 2018, str. 78 & 106).

4 PREGLED NAJPERSPEKTIVNEJŠIH GOSPODARSKIH PANOG ZA UPORABO TEHNOLOGIJE VERIŽENJA PODATKOVNIH BLOKOV

Pri pripravi tega članka smo ubrali podobno pot kot Andoni in drugi v njihovem članku z naslovom Tehnologija veriženja podatkovnih blokov v energetskem sektorju: Sistematičen pregled izzivov in priložnosti (angl. Blockchain technology in the energy sector: A systematic review of challenges and opportunities.) (Andoni in drugi, 2019, str. 143–156). Z metodo pregleda literature smo poizkusili združiti znanje iz znanstvenih člankov in drugih virov, kjer se združuje znanje s področja te tehnologije (poročila svetovalnih podjetij, blogi, bele knjige projektov itd.). Uvodnemu delu s predstavitevjo osnovnih principov delovanja tehnologije, ključnih prednosti in slabosti sledi pregled in ocena najperspektivnejših dejavnosti za uporabo te tehnologije. Do izbranih dejavnosti smo prišli po preučitvi vseh 21. področji dejavnosti, definiranih s Standardno klasifikacijo dejavnosti. Za posamezna področja smo pregledali osnovne možnosti in doslej predlagane načine uporabe te tehnologije, morebitno uporabo pa smo preučil tudi s pravnega, tehnološkega in sociološkega vidika. Glede na doslej znane primere uporabe, številčnost poizkusov ter ob upoštevanju prej navedenih prednosti in slabosti, smo nato ocenili najbolj obetavna področja uporabe. V nadaljevanju povzemamo ugotovitve za štiri področja, pri vsakem od področji pa je predstavljen tudi konkreten primer uporabe oz. poizkusa uporabe te tehnologije.

5 FINANČNE IN ZAVAROVALNIŠKE DEJAVNOSTI

Kot prvo lahko izpostavimo finančne in zavarovalniške dejavnosti, kamor spadajo med drugim bančne dejavnosti, dejavnosti finančnih skladov in holdinгов (Statistični urad Republike Slovenije, 2010, str. 237). V literaturi se področje finančnih dejavnosti omenja kot eno od najperspektivnejših za uporabo tehnologije veriženja podatkovnih blokov (The Economist, 2015). Prav tako se tu tudi največ vlaga v razvoj projektov uporabe te tehnologije. Subjekti iz ZDA, ki so povezani z finančnimi dejavnostmi, imajo največ vloženih zahtevkov za pridobitev patenta iz področja tehnologije veriženja podatkovnih blokov (Lee, 2018). Zelo veliko zanimanje izkazujejo banke, zagonska podjetja in strokovna javnost za uporabo te tehnologije v procesu mednarodnih denarnih trans-

ferjev med bankami. Proces prenosa denarja iz banke v eni državi v tujo banko namreč večinoma še vedno traja več dni. Čeprav so nekateri nacionalni in regionalni sistemi za poravnavo, kot je na primer Evropski sistem SEPA, precej učinkoviti, pa mednarodni sistem poravnave temelji na zapleteni mreži klirinških organov in korespondenčnih bank. V literaturi so prisotna pričakovanja, da bi lahko s tehnologijo veriženja podatkovnih blokov odpravili te probleme (Seffinga, Lyons & Bachmann, 2017, str. 3).

Ker je za finančne dejavnosti je značilna visoka stopnja reguliranosti, je pravni vidik eden ključnih pri ocenjevanju možnosti uporabe te tehnologije in tako ni presenečenje, da ti predpisi postavljajo kar nekaj ovir. Zakon o bančništvu (Ur. l. RS št. 25/15, 44/16 – ZRPPB, 77/16 – ZCKR, 41/17 in 77/18 – ZTFI-1) v 125. in 126. členu definira kot zaupne podatke vse podatke, dejstva in okoliščine o posamezni stranki, s katerimi razpolaga banka, in določa, da jih je banka dolžna varovati. To pomeni, da teh podatkov ni moč objaviti in uporabljati na javni verigi podatkovnih blokov. Pomembna je tudi Direktiva Evropske unije št. 2015/849 o preprečevanju uporabe finančnega sistema za pranje denarja ali financiranje terorizma. Ena od ključnih zahtev, ki jih bankam nalaga ta direktiva, je izvajanje ukrepov za poznavanje svojih strank (angl. know your customer – KYC), kar vključuje preverjanje istovetnosti strank in oceno tveganja za zlorabo bančnih računov za nezakonite namene. Zaradi varstva osebnih podatkov bo to na verigi podatkovnih blokov težko izvajati.

Pri informacijskih sistemih na področju finančnih storitev, še posebej bančništva in zavarovalništva, sta v ospredju zanesljivost in varnost sistemov. Ker se sistemi med seboj povezujejo, jih je zelo težko zamenjati z novejšimi. Tako je na primer zasnova najpomembnejšega medbančnega mednarodnega sistema Swift³, ki se uporablja pri okoli polovici vseh mednarodnih plačil na svetu, stara že 45 let (Arnold, 2018). Ker tehnologija veriženja podatkovnih blokov predstavlja veliko spremembo in drugačno arhitekturo glede na sedanje medbančne sisteme, bo klub interesu bank to tehnologijo zelo težko uveljaviti (Larios-Hernández, 2017, str. 870). Zaradi pravnih in tehničnih ovir bo potrebno razviti rešitve, ki bodo uporabljale drugačno arhitekturo in način delova-

³ <https://www.swift.com/our-solutions/global-financial-messaging> (zadnji obisk 10.9.2020)

nja, kot je postavljena v osnovni zasnovi delovanja omrežja in tehnologije veriženja podatkovnih blokov.

Eden od projektov uporabe te tehnologije na področju medbančnih transakcij je rešitev RippleNet⁴, razvita s strani ameriškega podjetja Ripple Labs Inc, ki želi vzpostaviti sistem, ki bi omogočal mednarodne transakcije med bankami s pomočjo kriptovalute XRP. Ta valuta bi opravljala vlogo nekakšne vmesne valute med dvema klasičnima valutama. Vendar pa je potrebno poudariti, da kljub prevladujočemu prepričanju, pri rešitvi RippleNet, strogo gledano ne gre za tehnologijo veriženja podatkovnih blokov temveč gre za tehnologijo razpršene evidence (angl. distributed ledger technology – DLT). Ta je sicer podobna in ima nekatere elemente in lastnosti tehnologije veriženja podatkovnih blokov, vendar predstavlja širši pojem oz. celoten sklop tehnologij. Pri tehnologiji razpršene evidence se zapisi shranjujejo en za drugim v rastočo evidenco in ne v verigo podatkovnih blokov. Vsak zapis pa mora biti potrjen s strani nabora potrjevalcev (Government Office for Science, 2016, str. 16-17). Omrežje RippleNet se tako kar precej razlikuje od klasičnega omrežja veriženja podatkovnih blokov, saj v omrežju ni rudarjev, transakcije pa se ne zapisujejo v podatkovne bloke, ki bi tvorili verigo. Transakcije preverja in potrjuje omejeno število izbranih in preverjenih vozlišč, gre za protokol imenovan Ripple Protocol Consensus Algorithm (RPCA). Transakcije pa se zapisujejo v razpršeno evidenco, kar pomeni, da vozlišča, vsako pri sebi, hranijo kopijo enega enotnega seznama transakcij- evidenco (Ripple, 2017, str. 4–16). Ta rešitev je tehnično manj zahtevna in ni povsem decentralizirana v primerjavi s tehnologijo veriženja podatkovnih blokov in je zaradi tega lažje izvedljiva in bolj verjetna. Podobne pristope uporabe tehnologije razpršenih evidenc sta ubrala tudi druga večja poizkusa uporabe tehnologije veriženja podatkovnih blokov v bančništvu. To sta konzorcija bank R3 z rešitvijo Corda in platforma Stellar (Newton, 2018). Najpomembnejša aplikacija podjetja Ripple Labs je xCurrent za izvajanje transakcij med bankami. Ta uporablja poseben sistem za pošiljanje sporočil med bankami, preko katerega se koordinira izmenjavo informacij in denarnih sredstev, imenovan *Interledger Protocol* ali *ILP*. xCurrent v primeru dveh bank iz različnih držav, ki uporabita korespondenčno banko za usmerjanje plačil, preko

sistema *ILP* pridobi podatke o provizijah in izračuna celotne stroške transakcije. Sledi validacija podatkov pred izvedbo same transakcije. Vse tri banke poznajo vse podatke in lahko potrdijo transakcijo vnaprej in tako zagotovijo visoko stopnjo uspešnosti transakcij. Nato sledi koordinacija sredstev med zasebnimi *ILP* glavnimi knjigami bank (npr. nostro računi). V vseh treh knjigah aplikacija xCurrent pripravi in zaklene sredstva potrebna za transakcijo. Vse *ILP* knjige nato generirajo elektronske podpise, ki zagotavljajo, da so sredstva za transakcijo na voljo. Nato se sredstva v vseh treh knjigah sočasno sprostijo. Tako napaka pri izvedeni poravnavi ni mogoča, transakcija se izvede v celoti ali pa se v primeru napake v celoti ne izvede (Ripple, 2017, str. 4–16).

Na področju zavarovalništva bi lahko tehnologijo veriženja podatkovnih blokov s pomočjo pametnih pogodb uporabili tako, da bi te spremljale ali so vsi pogoji za veljavnost zavarovanja izpolnjeni (npr. vse zavarovalniške premije so plačane), nato pa bi v primeru spremembe stanja predmeta zavarovanja avtomatsko izplačale zavarovalnino. Tu bi bilo ključno, da so preroki, ki poročajo v stanjih preverjeni, potrjeni in verodostojni (Tsankov, 2018). Primer poizkusa uporabe tehnologije veriženja podatkovnih blokov in pametnih pogodb je projekt Insuwave⁵ podjetij Ernst and Young in Guardtime, ki sta v sodelovanju z več logističnimi podjetji razvili platformo za poslovna zavarovanja s poudarkom na transportu. Obstajajo tudi ideje, da bi se tehnologijo veriženja podatkovnih blokov uporabljalo za hitrejšo preverjanje novih strank (predvsem za izmenjavo podatkov med različnimi zavarovalnicami) ter za preprečevanje zavarovalniških goljufij. To pa naj bi dosegli s pomočjo varnega hranjenja in deljenja informacij na verigi podatkovnih blokov (Lorenz in drugi, 2016, str. 3–5).

6 PROMET IN SKLADIŠČENJE

Med bolj perspektivne za uporabo tehnologije veriženja podatkovnih blokov spada tudi področje prometa in skladiščenja, torej logistike. Na tem področju je bolj zanimiv del povezan z prevozom blaga in zajema poslovanje med podjetji. Procesi v prometu in skladiščenju vključujejo večje število poslovnih partnerjev, ki se med seboj morda ne poznajo, poleg tega mednarodni promet blaga vključuje veliko količino administracije. In prav večje zaupanje in manj-

⁴ <https://ripple.com/ripple.net> (zadnji obisk 10.9.2020)

⁵ <https://insurwave.com/> (zadnji obisk 18.9.2020)

šo stopnjo administracije želijo na to področje uvesti projekti uporabe tehnologije veriženja podatkovnih blokov.

Mednarodni promet in skladiščenje blaga sta predmet urejanja številnih lokalnih zakonov, predpisov in dajatev. Čeprav bi rešitve na podlagi tehnologije veriženja podatkovnih blokov lahko povečale transparentnost in zmanjšale obseg administracije, pa ne bi bilo mogoče zaobiti vseh uvozno-izvoznih postopkov in dokumentov. Ne gre namreč pričakovati, da bi države povsem prilagodile zakonodajo tehnologiji veriženja podatkovnih blokov. Zaradi tega bodo morale biti rešitve na podlagi te tehnologije dovolj prilagodljive na vse posebnosti različnih trgov (Groenfeldt, 2017).

Kljub temu, da so pričakovanja o vplivu tehnologije veriženja podatkovnih blokov na področju prometa in skladiščenja dokaj velika, pa za sedaj na tem področju primanjkuje ekspertnega znanja (Dobrovnik, Herold, Fürst & Kummer, 2018, str. 1). Anketa iz leta 2017 med 152 strokovnjaki s področja logistike je pokazala, da velika večina pričakuje pozitiven vpliv te tehnologije na področje logistike, vendar večina ne ve natanko, kako bodo do teh sprememb prišli (Hackius & Petersen, 2017, str. 11–12).

Rešitve na podlagi te tehnologije za področje logistike obljublajo zmanjšanje števila interakcij med strankami v postopkih, vendar pa bi še vedno vključevale vso sedaj predpisano dokumentacijo. Verjetno najbolj konkreten in najdlje razvit primer uporabe te tehnologije na področju logistike je platforma TradeLens⁶, ki je nastala v sodelovanju podjetja Maersk, največjega kontejnerskega prevoznika na svetu, in podjetja IBM, ki je eno najbolj aktivnih tehnoloških podjetij na področju tehnologije veriženja podatkovnih blokov (Groenfeldt, 2017). TradeLens je rešitev za celostno digitalizacijo procesa dobave blaga, podprta s tehnologijo Hyperledger Fabric 1.0⁷ in računalniškimi oblakom IBM Cloud⁸ (White, 2018). Platforma Hyperledger, krovni projekt fundacije Linux, je odprt kodni sistem v prvi vrsti namenjen poslovanju med podjetji. Zaradi velike prilagodljivosti in modularnosti omogoča dobro prilagajanje različnim primerom uporabe. Tako se lahko uporabijo različne stopnje zasebnosti verige podatkovnih blokov. Podobno kot pri Ripplu pa tudi tu ne gre strogo gle-

dano za tehnologijo veriženja podatkovnih blokov temveč bolj za neko vrsto tehnologije razpršene evidence (DLT). Platforma TradeLens pri sledenju in upravljanju posamičnih pošilk deluje s pomočjo pametnih pogodb, ki se v platformi Hyperledger Fabric imenujejo tudi chaincode. Maersk je v letu 2018, ko je testna rešitev zaživela, uspel k projektu privabiti več kot 100 organizacij, kot so pristanišča, špediterji, ladijski prevozniki pa tudi carinski organi (Scott, 2018). V prvem letu testiranja platforme se je na njej zabeležilo kar 154 milijonov dogodkov povezanih z blagovnim prometom (Maersk, 2018).

7 TRGOVINA, VZDRŽEVANJE IN POPRAVILA MOTORNIH VOZIL

Trgovina, vzdrževanje in popravila motornih vozil je zelo široko področje, ki zajema trgovino na drobno in debelo vseh vrst blaga, v to področje pa spadajo tudi vse storitve povezane s prodajo. Tako v to področje spadajo tudi dejavnosti popravila motornih vozil (Statistični urad Republike Slovenije, 2010, str. 197). Trgovina je zadnji člen v dobavni verigi blaga do končnega kupca in se tako močno povezuje s področjem prevoza in skladiščenja. Za uporabo tehnologije veriženja podatkovnih blokov je zanimivo sledenje porekla blaga in poti po dobavni verigi. V to področje spada tudi trgovanje s hrano in zdravili pri katerih je, zaradi vpliva na zdravje ljudi, še posebej pomembno spremljanje njihovega porekla in poti od proizvodnje do končnega potrošnika.

Kot je bilo na že začetku nakazano, zapisi na verigi podatkovnih blokov ne predstavljajo pravno veljavnega dokaza o lastništvu nekega sredstva izven verige podatkovnih blokov. Temu primerno se ta tehnologija ne more uporabljati za trgovino samo, temveč jo je možno uporabiti kot platformo za sledenje zgodovine in stanja nekega sredstva (Reed, Sathyanarayan, Ruan & Collins, 2018, str. 170–181). Zanimiva je uporaba te tehnologije pri prodaji rabljenih vozil in beleženju stanja vozil tekom njihove življenjske dobe. Na verigi podatkovnih blokov bi tako lahko shranjevali podatke o zgodovini stanja in vzdrževanja vozila (Berryhill, Bourgerie & Hanson, 2018, str. 43). V povezavi s preprečevanjem zlorab na tem področju in s tem povezanim beleženjem zgodovine vozil velja poudariti, da Zakon o motornih vozilih (Ur. l. RS, št. 75/2017) manipulacijo s kilometrskim števcem prikaza prevožene razdalje vozila, opredeljuje kot kaznivo dejanje.

⁶ <https://www.tradelens.com/platform> (zadnji obisk 19.9.2020)

⁷ <https://www.hyperledger.org/hyperledger-fabric-1-0> (zadnji obisk 19.9.2020)

⁸ <https://www.ibm.com/cloud> (zadnji obisk 19.9.2020)

Leta 2016 je bila sprejeta uredba Evropske Komisije (2016/161), ki je določila podrobna pravila za zaščitne elemente na ovojnini zdravil za uporabo v humani medicini. Uredba od proizvajalcev zdravil zahteva t. i. serializacijo, kar pomeni, da je potrebno vsaki proizvedeni seriji zdravil dodeliti unikatno šifro in ustrezno označiti pakiranje izdelkov (šifra in črtna koda) (Ur. l. EU št. 2016/161). Že pred tem, leta 2013 je bil podoben zakon sprejet tudi v ZDA in sicer Zakon o varnosti oskrbe z zdravili, ki za implementacijo določa časovni okvir desetih let. Ta pravila naj bi zagotovila, da bi bilo mogoče zelo natančno slediti posameznim farmacevtskim izdelkom po celotni dobavni poti od proizvodnje do končnega kupca. To je področje, kjer se lahko v prihodnosti uporabi tehnologija veriženja podatkovnih blokov (Kherrat & Hernandez, 2018). Kar zadeva tehnični vidik serializacije zdravil omenjena Evropska direktiva nalaga, da je potrebno podatke o serijah zdravil pošiljati v centralni sistem regulatornih organov. Med tem ko, se je za tak pristop odločila tudi Kitajska, pa Ameriška zakonodaja ne predvideva centraliziranega pristopa, temveč naj bi proizvajalci podatke hranili sami. Slednje bi povzročilo veliko število podatkovnih baz in različnih rešitev, kar pomeni, da je potencial za uporabo tehnologije veriženja podatkovnih blokov na tem področju v ZDA večji kot v Evropi ali na Kitajskem (Whyte, 2016, str. 10–11).

Kot primer poizkusa uporabe tehnologije veriženja podatkovnih blokov za preprečevanje goljufij z manipulacijo števca prevoženih kilometrov v vozilih lahko navedemo skupni projekt enega največjih proizvajalcev avtomobilskih sestavnih delov, nemškega podjetja Bosch in univerze ETH Zürich. Rešitev vključuje namestitev majhne naprave v vozilo, ki zajema različne podatke o stanju vozila in jih najprej pošlje v računalnik ali pametni telefon lastnika avtomobila, kjer se neobdelani podatki šifrirani. Za identifikacijo bi uporabili identifikacijsko številko vozila (angl. vehicle identification number – VIN), vtisnjeno na šasiji vozila. Transakcija se lokalno podpisuje z zasebnim ključem uporabnika in šele nato se zgoščena vrednost podatkov pošlje preko interneta v verigo podatkovnih blokov, ki temeljijo na platformi Ethereum. Vsi podatki bi bili tako pred prenosom na verigo podatkovnih blokov lokalno šifrirani. Lastnik vozila bi lahko prostovoljno delil zapise o zgodovini prevoženih kilometrov s komurkoli bi želel, sistem pa bi zagotavljal, da se podatki ne morejo spreminja-

ti za nazaj (Chanson, Fleisch, Wortmann & Bogner, 2017, str. 2–4).

Tudi na področju sledenja zdravil se že pojavljajo določeni projekti, enega izmed njih razvija nemško tehnološko podjetje SAP. Rešitev bi podatke o serializaciji in sledenju poti zdravil po dobavni verigi shranjevala na verigo podatkovnih blokov zasnovano na platformi MultiChain⁹. Ta veriga bi bila zasebne tipa, vozlišča pa bi poleg SAP-a predstavljali tudi proizvajalci zdravil. Z zapisovanjem podatkov na verigo podatkovnih blokov bi se izognili številnim razdrobljenim podatkovnim bazam različnih proizvajalcev, trgovci in kupci pa bi lahko na verigi preverili pristnost zdravila. Rešitev je za sedaj še v fazi razvoja (Morris, 2018a).

8 INFORMACIJSKE IN KOMUNIKACIJSKE DEJAVNOSTI

Področje informacijske in komunikacijskih dejavnosti je precej široko področje, ki vključuje založništvo (knjig, glasbe, filmov), telekomunikacije, obdelavo podatkov, izdajanje programja (Statistični urad Republike Slovenije, 2010, str. 230–236). Tehnologija veriženja podatkovnih blokov je na tem področju najbolj zanimiva za morebitno zapisovanje in nadziranje uporabe licenc programske opreme (Felin & Lakhani, 2018, str. 35).

Za nadzor in upravljanje z licencami programske opreme nekatera podjetja uporabljajo t. i. orodja za upravljanje s programsko opremo (angl. software asset management – SAM), ki pa so primarno namenjena za notranji nadzor in obvladovanje licenc s strani poslovnih uporabnikov. Med tem pa se morebitne rešitve na osnovi tehnologije veriženja podatkovnih blokov bolj osredotočajo na nadzor s strani ponudnikov programske opreme. Z uporabo te tehnologije bi lahko izboljšali ažurnost informacij o uporabi programov, ki zaradi posrednikov, kot so velike distribucijske platforme, ni najboljša (Morris, 2018b).

Zanimiva je rešitev na osnovi tehnologije veriženja podatkovnih blokov za upravljanje licenc računalniških igrice, ki jo je v sodelovanju s svetovalnim podjetjem Ernst and Young razvilo podjetje Microsoft (Ernst & Young Global Limited, 2018). Rešitev temelji na verigi podatkovnih blokov Quorum¹⁰ in Microsoftovi oblaci rešitvi Azure¹¹ (Varshney,

⁹ <https://www.multichain.com/> (zadnji obisk 20.9.2020)

¹⁰ <https://consensus.net/quorum/> (zadnji obisk 22.9.2020)

¹¹ <https://azure.microsoft.com/> (zadnji obisk 22.9.2020)

2018), ter bi v verigi podatkovnih blokov implementirala pametne pogodbe. S pomočjo te rešitve bi razvijalci in založniki računalniških iger, ki svoje igrice ponujajo na Microsoftovi platformi Xbox Live, imeli v realnem času vpogled v prodajo njihovih izdelkov. Microsoftovi partnerji sicer dobivajo poročila o prodaji, a njihova priprava lahko traja dalj časa, tudi do 45 dni ali več (Ernst & Young Global Limited, 2018).

9 DRUGE POGOSTO OMENJENE DEJAVNOSTI

Možnost uporabe tehnologije veriženja podatkovnih blokov se sicer omenja v povezavi z domala vsemi področji dejavnosti. Eno izmed največkrat omenjenih področji je zdravstvo in socialno varstvo. Vendar pa bi bilo zaradi zelo visokih standardov pri varstvu osebnih podatkov na področju zdravstva moč tehnologijo veriženja podatkovnih blokov možno uporabiti le kot varnostno plast za dostop do zdravniških evidenc in ne za shranjevanje zdravstvenih podatkov samih (Zhu, Wu, Gai & Choo, 2019, str. 529). Podobno velja za prav tako pogosto omenjeno področje javne uprave, kjer je urejanje javnih evidenc najbolj perspektivno področje (npr. zemljiški register). Vendar pa tudi tu obstajajo podobne zakonske ovire kot pri zdravstvu, hkrati pa ne rešujejo problema netočnih osnovnih, že obstoječih, podatkov (Vos, 2017, str. 23). Uporaba tehnologije veriženja podatkovnih blokov bi bila možna tudi na področju iger na srečo, kjer bi lahko preko pametnih pogodb in decentraliziranih aplikacij, t. i. DApps, uporabili omrežje verige podatkovnih blokov za sklepanje stav neposredno med stavci ali igralniško aplikacijo, ki bi bila posledično povsem decentralizirana (Gainsbury & Blaszczyński, 2017, str. 483). Ena izmed slednjih je na primer porazdeljena aplikacija Dice¹² deluje na omrežju verige podatkovnih blokov ESO¹³. Med glavnimi izzivi tovrstne uporabe so ponovno zakonske omejitve, saj zakonodaja na področju iger na srečo v večini držav zahteva implementacijo že omenjenega koncepta poznavanja svoje stranke in ukrepov proti preprečevanju pranja denarja (Gainsbury & Blaszczyński, 2017, str. 485).

10 SKLEP

Tehnologija veriženja podatkovnih blokov se precej razlikuje od vseh zdaj uveljavljenih tehnologij in je unikatna v tem, da omogoča vzpostavitev zaupanja

v decentraliziranem okolju med strankami, ki se ne poznajo in si med seboj ne zaupajo. V članku smo na podlagi raziskovalne metode pregleda literature, izmed vseh 21 področij dejavnosti, definiranih v Standardni klasifikaciji dejavnosti, določili najbolj obetavne dejavnosti za uporabo te tehnologije. Perspektivnost panoge za uporabo te tehnologije smo ocenili na podlagi predlogov uporabe v posamezni dejavnosti ter pregledu pravnega, tehnološkega in sociološkega vidika morebitne implementacije te tehnologije na posameznem področju. Pri ocenjevanju smo upoštevali tudi glavne prednosti in slabosti ter njihovo težo v posamičnih primerih uporabe.

V članku smo podrobneje predstavili le najbolj obetavna področja dejavnosti za uporabo tehnologije veriženja podatkovnih blokov. Tako smo izpostavili finančne in zavarovalniške dejavnosti, kjer je najbolj perspektiven proces mednarodnih denarnih transferjev. V tem primeru je sicer bolj verjetna uporaba tehnologije razpršenih evidenc kakor strogo definirane tehnologije veriženja podatkovnih blokov. Med bolj obetavne smo uvrstili tudi področje prometa in skladiščenje, kjer bi se lahko uporabljalo to tehnologijo za boljšo izmenjavo podatkov pri logističnih procesih, za zmanjšanje števila stikov med vpletenimi poslovnimi partnerji in povečanje zaupanja v pravilnost podatkov. Podobno smo ugotovili, da ima potencial tudi področje trgovine, vzdrževanja in popravila motornih vozil, ki se v delu, ki se nanaša na trgovanje s hrano povezuje tudi s področjem kmetijstva, lova, gozdarstva in ribištva. Na tem področju bi ta tehnologija lahko omogočila boljše spremljanje in zanesljivejše hranjenje podatkov o zgodovini blaga. Kot panogo, kjer je verjetna uporaba tehnologije veriženja podatkovnih blokov, smo identificirali tudi področje informacijskih in komunikacijskih dejavnosti, kjer bi se lahko ta tehnologija uporabljala za hranjenje in nadzor licenc programja.

Druge dejavnosti izkazujejo manjšo verjetnost za uporabo te tehnologije. Za večino dejavnosti predvsem ne obstaja izkazana dovolj velika potreba po decentraliziranem zaupanju ali pa so ostale omejitve, kjer prednjačijo omejitve pri razširljivosti ter zakonske omejitve, prevelike. Ocenjujemo, da je tehnologija veriženja podatkovnih blokov bolj smiselno povezovati s specifičnimi primeri uporabe in ne toliko s celotnimi področji dejavnosti. Za uporabo so najbolj primerne rešitve, kjer ni obdelave osebnih podatkov ali lastniških in drugih pravic na sredstvih, temveč se

¹² <https://dice.one/portfolio> (zadnji obisk 3.10.2020)

¹³ <https://eos.io/eos-public-blockchain/> (zadnji obisk 3.10.2020)

tehnologija veriženja podatkovnih blokov uporablja bolj kot dokaz o stanju in zgodovini nekega sredstva.

Ker se veliko projektov ukvarja z razvojem tehnologij razpršenih evidenc, in ne le strogo tehnologije veriženja podatkovnih blokov, menimo, da bi se bilo smiselno posvetiti tudi raziskovanju ostalih različic te tehnologije.

LITERATURA

- [1] Alharby, M & Moorsel, A. (2017, avgust). *Blockchain-Based Smart Contracts: A Systematic Mapping Study*. Pridobljeno 20. decembra 2018 iz https://www.researchgate.net/publication/319603816_Blockchain_Based_Smart_Contracts_A_Systematic_Mapping_Study
- [2] Allen, M. (2018, 14. december). *Switzerland sets legal foundations for blockchain industry*. Pridobljeno 10. februarja 2019 iz https://www.swissinfo.ch/eng/business/dlt-report_switzerland-sets-legal-foundations-for-blockchain-industry/44617654
- [3] Arnold, M. (2018, 6. junij). *Ripple and Swift slug it out over cross-border payments*. Pridobljeno 16. decembra 2018 iz <https://www.ft.com/content/631af8cc-47cc-11e8-8c77-ff51caedcde6>
- [4] Bacon, J., Michels, J. D., Millard, C. & Singh, J. (2018). *Blockchain Demystified: A Technical and Legal Introduction to Distributed and Centralised Ledgers*. *Richmond Journal of Law & Technology*, 25(1), 1-106.
- [5] Bambara, J. J. & Allen, P. (2018). *Blockchain A Practical Guide to Developing Business, Law, and Technology Solutions*. New York City: McGraw-Hill Education
- [6] Bank for International Settlements. (2018, junij). *Annual Economic June 2018 Report*. Pridobljeno 28. decembra 2018 iz <https://www.bis.org/publ/arpdf/ar2018e.pdf>
- [7] Berryhill, J., Bourgerly, T. & Hanson, A. (2018). *Blockchains Unchained: Blockchain Technology and its Use in the Public Sector*. *OECD Working Papers on Public Governance*, 28.
- [8] Bitcoin.com. (2019). *Blockchain Size*. Pridobljeno 15. novembra 2019 iz <https://charts.bitcoin.com/btc/chart/blockchain-size>
- [9] Chanson, M., Fleisch, M., Wortmann, F. & Bogner, A. (2017, avgust). *Blockchain as a Privacy Enabler: An Odometer Fraud Prevention System*. Pridobljeno 17. novembra 2018 iz http://cocoa.ethz.ch/media/documents/2017/08/None_UbiComp_2017-Privacy_Poster_final_1.pdf
- [10] ConsenSys (2019, 10. januar). *The Thirddening: What You Need To Know*. Pridobljeno 10. avgusta 2020 iz <https://media.consensys.net/the-thirddening-what-you-need-to-know-df96599ad857>
- [11] Dobrovnik, M., Herold, D. M., Fürst, E. & Kummer, S. (2018, 3. september). *Blockchain for and in Logistics: What to Adopt and Where to Start*. *Logistics 2018*, 2(3).
- [12] Dobson, D. (2018, 13. februar). *The 4 Types of Blockchain Networks Explained*. Pridobljeno 10. septembra 2020 iz <https://iltanet.org/blogs/deborah-dobson/2018/02/13/the-4-types-of-blockchain-networks-explained?ssopc=1>
- [13] Ernst & Young Limited. (2018, 21. junij). *EY and Microsoft launch blockchain solution for content rights and royalties management for media and entertainment industry*. Pridobljeno 15. decembra 2018 iz https://www.ey.com/en_gl/news/2018/06/ey-and-microsoft-launch-blockchain-solution-for-content-rights
- [14] Ethereum Foundation (2020, 11. junij). *Proof of Stake FAQs*. Pridobljeno 25. september 2020 iz <https://eth.wiki/en/concepts/proof-of-stake-faqs>
- [15] Ethereum Foundation (2020, 28. september). *Ethereum 2.0 (Eth2)*. Pridobljeno 1. oktobra 2020 iz <https://ethereum.org/en/eth2/>
- [16] Felin, T. & Lakhani, K. (2018). *What Problems Will You Solve With Blockchain?* *MIT Sloan Management Review*, 60, 32–38.
- [17] Gainsbury, M. & Blaszczynski, A. (2017, september). *How Blockchain and Cryptocurrency Technology Could Revolutionize Online Gambling*. *Gaming Law Review*, 21(7), 482–492.
- [18] Government Office for Science. (2016). *Distributed Ledger Technology: Beyond Block Chain*. Pridobljeno 1. junija 2018 iz https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf
- [19] Groenfeldt, T. (2017, 5. marec). *IBM And Maersk Apply Blockchain To Container Shipping*. Pridobljeno 9. novembra 2018 iz <https://www.forbes.com/sites/tomgroenfeldt/2017/03/05/ibm-and-maersk-apply-blockchain-to-container-shipping/#1329a0c03f05>
- [20] Hackius, N. & Petersen, M. (2017, oktober). *Blockchain in Logistics and Supply Chain: Trick or Treat?* Pridobljeno 11. novembra 2018 iz https://tubdok.tub.tuhh.de/bitstream/11420/1447/1/petersen_hackius_blockchain_in_scm_and_logistics_hicl_2017.pdf
- [21] Hayes, A. (2020, 24. septembra). *Proof of Capacity (Cryptocurrency)*. Pridobljeno 7. septembra 2020 iz [https://www.investopedia.com/terms/p/proof-capacity-cryptocurrency.asp#:~:text=Proof%20of%20capacity%20\(PoC\)%20is,mining%20rights%20and%20validate%20transactions](https://www.investopedia.com/terms/p/proof-capacity-cryptocurrency.asp#:~:text=Proof%20of%20capacity%20(PoC)%20is,mining%20rights%20and%20validate%20transactions)
- [22] Hooper, M. (2018, 22. februar). *Top five blockchain benefits transforming your industry*. Pridobljeno 26. november 2018 iz <https://www.ibm.com/blogs/blockchain/2018/02/top-five-blockchain-benefits-transforming-your-industry>
- [23] Karantias, K., Kiayias, A., & Zindros, D. (2020, februar). *Proof-of-burn*. *International Conference on Financial Cryptography and Data Security*, 24, 523–540.
- [24] Kherrat, R. & Hernandez, A. (2018, 24. oktober). *Understanding the complexities of global serialisation & traceability regulations*. Pridobljeno 22. decembra 2018 iz <https://www.europeanpharmaceuticalreview.com/webinar/79102/understanding-the-complexities-of-global-serialisation-traceability-regulations>
- [25] Lai, R. & Lee Kuo Chuen, D. (2018). *Blockchain- From Public to Private*. *Handbook of Blockchain, Digital Finance, and Inclusion*, 2, 145–177.
- [26] Larios-Hernández, G. J. (2017, november-december). *Blockchain entrepreneurship opportunity in the practices of the unbanked*. *Business Horizons*. 60(6), 865–874.
- [27] Lee, A. (2018, 12. januar). *Blockchain patent filings dominated by financial services industry*. Pridobljeno 14. novembra 2019 iz <http://patentvue.com/2018/01/12/blockchain-patent-filings-dominated-by-financial-services-industry>
- [28] Lorenz, J., Münstermann, B., Higginson, M., Olesen, P. B., Bohlken, N. & Ricciardi, V. (2016, julij). *Blockchain in insurance –opportunity or threat?* Pridobljeno 18. decembra 2019 iz <https://www.mckinsey.com/~/media/mckinsey/industries/financial%20services/our%20insights/blockchain%20in%20insurance%20opportunity%20or%20threat/blockchain-in-insurance-opportunity-or-threat.ashx>
- [29] Maersk. (2018, 9. avgust). *Maersk and IBM Introduce TradeLens Blockchain Shipping Solution*. Pridobljeno 9. decembra 2018 iz <https://www.maersk.com/en/news/2018/06/29/maersk-and-ibm-introduce-tradelens-blockchain-shipping-solution>

- [30] Millard, C. (2018, avgust). Blockchain and law: Incompatible codes? *Computer Law & Security Review*. 34(4), 843–846.
- [31] Morris, N. (2018a, junij). *SAP leads Pharma Supply Chain blockchain*. Pridobljeno 9. januarja 2019 iz <https://www.ledgerinsights.com/sap-pharma-supply-chain/>
- [32] Morris, N. (2018b, junij). *EY and Microsoft announce royalties blockchain*. Pridobljeno 14. decembra 2018 iz <https://www.ledgerinsights.com/ey-microsoft-royalties-blockchain/>
- [33] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Najdeno 9. junija 2018 na spletnem naslovu <https://bitcoin.org/bitcoin.pdf>
- [34] Newton, D. (2018, 5. junij). *What is Corda?*. Pridobljeno 13. decembra 2018 iz <https://medium.com/corda/what-is-corda-6417b14c8dc7>
- [35] Ramuta, M. (2018, 30. april). *Why DPoS is not really a PoS, but rather a PoA protocol*. Pridobljeno 6. septembra 2020 iz <https://hackernoon.com/why-dpos-is-not-really-a-pos-but-rather-a-poa-protocol-5bb1aa305625>
- [36] Reed, C., Sathyanarayan U. M., Ruan, S. & Collins, J. (2018). Beyond BitCoin- legal impurities and off-chain assets. *International Journal of Law and Information Technology*, 26(2), 160–182.
- [37] Reyna, A., Martín, C., Chen, J., Soler, E. & Díaz, M. (2018). On blockchain and its integration with IoT. *Challenges and opportunities. Future Generation Computer Systems*, 88, 173–190.
- [38] Ripple. (2017, oktober). *Product Overview*. Pridobljeno 20. decembra 2018 iz https://ripple.com/files/ripple_product_overview.pdf
- [39] Scott, T. (2018, 27. september). *TradeLens: How IBM and Maersk Are Sharing Blockchain to Build a Global Trade Platform*. Pridobljeno 9. decembra 2018 iz <https://www.ibm.com/blogs/think/2018/11/tradelens-how-ibm-and-maersk-are-sharing-blockchain-to-build-a-global-trade-platform/>
- [40] Seffinga, J., Lyons, L. & Bachmann, A. (2017). *The Blockchain (R)evolution – The Swiss Perspective*. Pridobljeno 15. aprila 2018 iz <https://www2.deloitte.com/content/dam/Deloitte/ch/Documents/innovation/ch-en-innovation-blockchain-revolution.pdf>
- [41] Statistični urad Republike Slovenije. (2010). *Standardna klasifikacija dejavnosti 2008*. Ljubljana: Statistični urad Republike Slovenije, 11.
- [42] The Economist. (2015, 31. oktober). *Blockchains: The great chain of being sure about things*. Pridobljeno 15. septembra 2018 iz <https://www.economist.com/briefing/2015/10/31/the-great-chain-of-being-sure-about-things>
- [43] Tsankov, A. (2018, 21. junij). *The “Oracle Problem» isn’t a Problem, and why Smart Contracts makes Insurance better for everyone*. Pridobljeno 1. oktobra 2020 iz <https://medium.com/@antsankov/the-oracle-problem-isnt-a-problem-and-why-smart-contracts-makes-insurance-better-for-everyone-8c979f09851c>
- [44] Varshney, N. (2018, 21. junij). *Microsoft launches ambitious blockchain project to help creators get paid*. Pridobljeno 15. decembra 2018 iz <https://thenextweb.com/hardfork/2018/06/21/microsoft-and-ey-launch-blockchain-for-copyrights-and-royalties/>
- [45] Vos, J. (2017, 10. februar). *Blockchain-Based Land Registry: Panacea, Illusion Or Something In Between? European Land Registry Association Annual Publication*, 7. Pridobljeno 3. januarja 2019 iz <https://www.elra.eu/wp-content/uploads/2017/02/10.-Jacques-Vos-Blockchain-based-Land-Registry.pdf>
- [46] White, M., (2018, 16. januar). *Digitizing Global Trade with Maersk and IB*. Pridobljeno 9. decembra 2018 iz <https://www.ibm.com/blogs/blockchain/2018/01/digitizing-global-trade-maersk-ibm/>
- [47] Whyte, J. (2016) *Serialization: An Implementation Guide*. Pridobljeno 9. januarja 2019 iz http://www.worldpharmatoday.com/whitepapers/rockwell_automation_11082016.pdf
- [48] Zhang, F., Cecchetti, E., Croman, K., Juels, A. & Shi, E. (2016). *Town Crier: An Authenticated Data Feed for Smart Contracts. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (str. 107–283). Dunaj. ACM.
- [49] Zhu, L., Wu, Y., Gai, K. & Choo K. R. (2019, februar). *Controllable and Trustworthy Blockchain-Based Cloud Data Management. Future Generation Computer Systems*, 91, 527–5

■

Bor Krizmanič je diplomiral in magistriral na Ekonomski fakulteti Univerze v Ljubljani, smer poslovna informatika. Ukvarjal se je z revizijo informacijskih sistemov in svetovanjem na področju informatizacije bank in zavarovalnic. Sedaj pa je asistent na Ekonomski fakulteti Univerze v Ljubljani, na katedri za poslovno informatiko in logistiko.

■

Aleš Groznik je redni profesor na Ekonomski fakulteti Univerze v Ljubljani, katedra za poslovno informatiko in logistiko. Glavna področja njegovega raziskovalnega dela predstavljajo management oskrbovalne verige, elektronsko poslovanje in sodobna mobilnosti.

Z umetno inteligenco podprt razvoj programske opreme

Mitja Gradišnik, Tina Beranič, Sašo Karakatič

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor
mitja.gradisnik@um.si; tina.beranic@um.si; saso.karakatic@um.si

Izvleček

Številni izzivi, na katere naletimo pri razvoju programskih rešitev, nas silijo, da neprestano iščemo nove pristope in prakse, s katerimi bi IT projekte realizirali boljše, hitreje in predvsem z nižjimi stroški. Želja po hitri in cenovno ugodni realizaciji IT projektov, višji stopnji njihove kakovosti ter nenazadnje v zadnjem času že kroničnem pomanjkanju usposobljenih IT strokovnjakov, so samo nekateri izmed izzivov, s katerimi se srečujemo v programskem inženirstvu. Pri naslavljanju omenjenih izzivov si v zadnjem času veliko obetamo od vpeljave umetne inteligence v proces razvoja programske opreme. Možnosti se kažejo predvsem v vpeljavi z naprednimi metodami umetne inteligence podprtih orodij, ki razvojno skupino razvijalcev aktivno podpirajo pri razvoju. Z umetno inteligenco podprta orodja odpirajo vrata odmiku od avtomatizacije ponavljajočih se trivialnih opravil in obljublajo možnost avtomatizacije intelektualno zahtevnejših in kompleksnih opravil, kar bi občutno razbremenilo razvijalce informacijskih rešitev.

Ključne besede: proces razvoja programske opreme, orodja razvoja programske opreme, umetna inteligenca

Abstract

In software engineering, many of the challenges we face during software development force us to constantly look for new approaches and practices to help deliver better and faster IT projects at lower costs. The desire for a fast and affordable implementation of IT projects, a higher level of quality of solutions and the chronic shortage of skilled IT developers are just some of the challenges we face. Recently, in addressing these challenges, the introduction of artificial intelligence into the software development process has shown a lot of promise especially in the form of advanced artificial intelligence-supported tools that actively support the development team in the development process. Artificial intelligence-enabled tools offer abilities beyond the automation of trivial repetitive tasks. Primarily, they introduce the automation of more complex tasks, the execution of which was exclusively in the domain of skilled IT professionals until recently.

Keywords: Software development process, software development tools, artificial intelligence, intelligent assistants.

1 UVOD

Podjetja, ki delujejo v sodobnem poslovnem okolju, se morajo skozi proces digitalne preobrazbe neprestano prilagoditi spremenjenim pričakovanjem digitalno vedno bolj ozaveščenih strank. Te postajajo iz leta v leto bolj naklonjene interakciji preko digitalnih medijev, vse pogosteje posegajo po spletnih nakupih, prav tako dostopajo do vrste storitev preko digitalnih kanalov (Cuesta, Ruesta, Tuesta, & Urbiola, 2015). Za številne banke po svetu postaja spletno in mobilno bančništvo vsaj toliko pomembno, če že ne pomembnejše, kot poslovanje preko mreže lastnih izpostav in bančnih avtomatov (Ross & Srinivas, 2018).

Pravočasno in ustrezno digitalno preoblikovanje podjetij postaja posledično eden izmed ključnih strateških ciljev, kateremu so podjetja v zadnjih letih predana. Nepravočasno prilagajanje potrebam in pričakovanjem strank bi se lahko tako hitro prelilo v zamujene poslovne priložnosti. Kljub občutnemu premiku v smeri digitalizacije storitev opravljenem v zadnjih letih, gospodarstvo še zmeraj ne izkorišča polnega potenciala, ki ga prinaša vpeljava sodobnih informacijsko-komunikacijskih tehnologij. Po trenutnih ocenah izkoriščajo Združene države Amerike 18 % potenciala, ki ga prinaša digitalizacija, evropske države v povprečju le 12 % (Bughin idr., 2016). Di-

gitalna preobrazba podjetja prav tako naslavlja procese notranje optimizacije delovanja podjetja, v okviru katerega se morajo podjetja priučiti veščin, kako izkoristiti množico razpoložljivih podatkov ter na kakšen način bi te podatke s pomočjo sodobnih informacijsko-komunikacijskih rešitev uporabiti, da bi z njihovo pomočjo bolje razumeli potrebe strank, trgov, konkurence in aktualne trende na trgu (Cagle, 2019).

Četudi podjetja uspejo pripraviti strategije digitalne preobrazbe, ki zmorejo učinkovito nasloviti aktualne potrebe na trgu, pogosto naletijo na izzive pri njeni realizaciji. Izkoriščanje potencialov digitalne preobrazbe čedalje bolj ovira pomanjkanje ustrezno usposobljenih kadrov inženirskih smeri, ki so ključni za uspešno realizacijo tovrstnih IT projektov. Kot kažejo raziskave analitičnih hiš Gartner (Griffin, Mok, Struckman, & Berry, 2018) in Deloitte (Giffi idr., 2018), usposabljanje potrebnih IT kadrov trenutno ne sledi potrebam gospodarstva. Problem pomanjkanja ustrezno usposobljenega inženirskega kadra je sicer mogoče zaznati dalj časa, projekcije trendov pa nakazujejo, da se bo stanje z leti še dodatno poslabšalo (Deming & Noray, 2018). Na izzive pomanjkanja usposobljenega inženirskega kadra je sicer potrebno gledati kot na širši družbeni problem, katerega morajo ustrezno nasloviti tako državni organi preko izobraževalnih programov kot privatni sektor s spodbujanjem vseživljenjskega učenja zaposlenih. A vendar enostavnih rešitev, ki bi čez noč odpravile težave pomanjkanja usposobljenega IT kadra, kratkoročno ne gre pričakovati. Ključni izziv tako ostaja, kako z obstoječimi kadri opraviti večjo količino dela.

V zadnjih letih si pri reševanju predstavljene problematike posebej veliko obetamo od razvoja tehnologij umetne inteligence. Te so v zadnjih letih dosegle zrelostni nivo, pri katerem tehnologije niso uporabne zgolj v izoliranih laboratorijskih okoljih, temveč jih je mogoče aplicirati v konkretne primere iz realnega življenja. Z razvojnimi orodji podprtimi s tehnologijami umetne inteligence je mogoče uspešno nasloviti aktualne probleme, ki spremljajo proces razvoja programske opreme. Tega namreč v veliki meri sestavljajo intelektualno zahtevna opravila, ki v veliki meri terjajo od razvijalcev mentalni napor. Z razvojnimi orodji podprtimi z metodami umetne inteligence bi lahko presegli nivo trenutne avtomatizacije mentalno nezahtevnih in pogosto ponavljajočih se opravil. V razvojni proces IT bi tako vpeljali višji nivo avtomatizacije, s katero bi bilo mogoče razvijalce pri razvoju IT rešitev razbremeniti tudi intelektu-

alno zahtevnejših opravil. Slednje bi imelo za posledico učinkovitejši in bolj optimalni razvojni proces informacijskih rešitev, s katerim bi lažje naslavljali izzive, ki jih prinaša pospešena digitalna transformacija podjetji, kateri smo priča v zadnjih letih.

2 UMETNA INTELIGENCA V ORODJIH ZA RAZVOJ PROGRAMSKIH REŠITEV

2.1 Pristopi in metode umetne inteligence

Umetna inteligenca je v zadnjih letih na polno vstopila v naša vsakodnevna življenja. Napovedani avtonomni avtomobili in pametni asistenti, kot so Siri, Alexa in Google, so zgolj najvidnejši predstavniki tehnologij, ki jih najpogosteje povezujemo z umetno inteligenco. Poleg navedenih izpostavljenih primerov je mogoče umetno inteligenco aplicirati na številna druga področja. Termin umetna inteligenca je sicer uporablja zgolj krovni in zelo posplošen izraz, pod katerim je združenih več družin med seboj zelo različnih pristopov in metod, ki so namenjeni reševanju najrazličnejših vsakodnevnih izzivov (Gradišnik, Karakatič, Mauša, Beranič, & Heričko, 2019).

Na področju programskega inženirstva je mogoče vključiti najrazličnejše metode umetne inteligence in jih uporabiti za reševanje širokega spektra aktualnih izzivov znotraj razvojnega cikla programske opreme. Področje umetne inteligence aktualno za programsko inženirstvo v grobem zajema različne pristope stojnega in globokega učenja, predstavitve znanja z grafi, semantičnimi mrežami in poslovnimi pravili, obdelavi naravnega jezika in avtomatizirani razpoznavi elementov v slikovnem ter avdio in video materialu (Lo Giudice, 2016). V bližnji prihodnosti se pričakuje se, da bo kombinacija navedenih metod umetne inteligence imela močan vpliv na vse stopnje razvojnega cikla programske opreme, in sicer tako, da bo razvijalcem omogočen razvoj boljše programske opreme v krajšem času (Lo Giudice, 2016). Metode umetne inteligence se na področje programskega inženirstva aplicirajo preko naprednih razvojnih orodij, za katera je značilno, da imajo karakteristike intelektualnega procesa značilnega za ljudi, kot je na primer zmožnost sklepanja, odkrivanja pomena, generalizacije in učenja iz preteklih izkušenj (Copeland, 2019).

2.2 Repozitoriji projektov programskih produktov

Kljub čedalje večji raznovrstnosti in dovršenosti algoritmov umetne inteligence ti sami po sebi niso

dovolj, da bi z njimi uspešno reševali predhodno izpostavljene izzive programskega inženirstva. Da izdelamo z umetno inteligenco podprta razvojna orodja, so poleg dodelanih algoritmov umetne inteligence potrebna zbirka kakovostnih učnih podatkov, nad katerimi je algoritme umetne inteligence mogoče učinkovito učiti. V praksi se dostopni repozitoriji programskih projektov, kot so na primer GitHub («GitHub», 2019), GitLab («GitLab», 2019), Bitbucket («Bitbucket», 2019a) in SourceForge («SourceForge», 2019), izkažejo za neprecenljiv vir znanja s področja programskega inženirstva. Repozitoriji programskih projektov so v osnovi virtualna okolja, znotraj katerih razvojne skupine v okviru aktivnosti razvoja programske opreme soustvarjajo raznovrstne artefakte potrebne za realizacijo programskih produktov (Güemes-Peña, López-Nozal, Marticorena-Sánchez, & Maudes-Raedo, 2018). Repozitoriji programskih projektov so po več desetletij splošne uporabe prerasli v ogromne zbirke najrazličnejših programskih projektov, ki jih lahko uporabimo kot temelj za izgradnjo s strojnim učenjem podprtih razvojnih orodij.

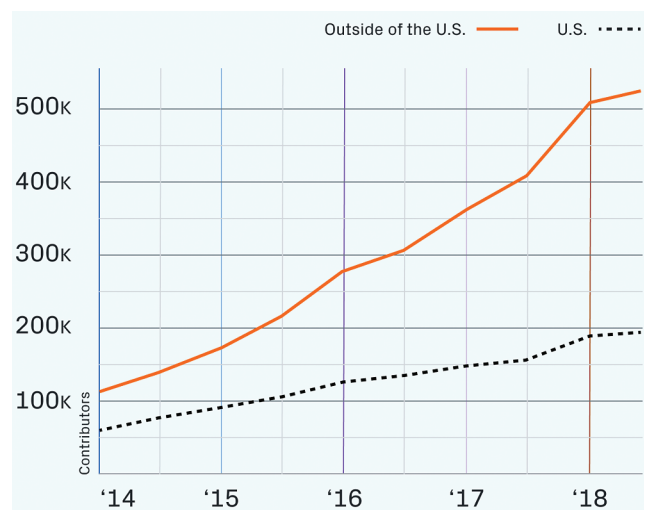
Poleg repozitorijev programskih projektov so pomemben vir domenskega znanja prav gotovo tudi spletne skupnosti za pomoč med razvijalci. Pri vidnejših predstavnikih tovrstnih skupnosti zagotovo ne moremo mimo spletne skupnosti razvijalcev StackOverflow («StackOverflow», 2019). Ta namreč ponuja ogromno bazo relevantnih razvojnih izzivov in njihovih rešitev, ki jih je mogoče učinkovito uporabiti pri izgradnji modelov v strojnem učenju.

Kot odziv na možnosti, ki jih prinaša analiza navedenih spletnih skupnosti in orodij, je vzniknila relativno mlada raziskovalna veja rudarjenja programskih repozitorijev (angl. Software Repository Mining), ki se ukvarja s sistematičnimi pristopi pridobivanja in analiziranja podatkov dostopnih v programskih repozitorijih (Kaur, Kaur, Chopra, & Kaur, 2020). V ospredju rudarjenja programskih repozitorijev ni zgolj izvorna koda programskih projektov, temveč vsi artefakti, ki nastajajo v okviru procesa razvoja programske opreme. Torej tudi zapisi v repozitoriju za sledenje programskim hroščem, sistemu za verzioniranje programske kode in različnih kanalih komunikacije med razvijalci, na primer v dopisnih seznamih (angl. mailing lists). Na podlagi relevantnih podatkov, izluščenih iz repozitorijev programskih projektov je mogoče razpoložljive algoritme umetne inteligence naučiti, da so ti zmožni samostoj-

nega reševanja problemov, na katere je moč naleteti med razvojem programske opreme, kar pa predstavlja temelj novodobnih s strani umetne inteligence podprtih razvojnih orodij.

O možnostih, ki jih rudarjenje repozitorijev projektov programskih produktov za razvoj inteligentnih razvojnih orodij prinaša, zgovorno povedo številke o uporabnikih in gostujočih repozitorijih, ki so jih uporabniki tovrstnih skupnosti ustvarili v zadnjih desetletjih. GitHub, z naskokom največji med navedenimi, je v avgustu 2019 poročal o več kot 50 milijonih uporabnikov, ki se učijo, si med seboj delijo in soustvarjajo več kot 100 milijonov repozitorijev programskih produktov (GitHub, 2020a, 2020b). Intenzivna je tudi rast uporabnikov, ki prispevajo programsko kodo. Graf rasti uporabnikov sicer med leti 2014–2018 prikazuje Slika 1. Za razliko, nekoliko manj uporabnikov beležijo pri konkurenčnih spletnih skupnostih Bitbucket. Njihova oblachna storitev je v aprilu 2019 poročala o 10 milijonih registriranih uporabnikov, ki so takrat upravljali več kot 28 milijonov repozitorijev projektov programskih produktov (Bitbucket, 2019b).

Programski projekti, ki se nahajajo v omenjenih repozitorijih, so strukturno izredno heterogeni. Temeljijo namreč na različnih programskih jezikih, uporabljajo najrazličnejša razvojna ogrodja in so spisana za široko paleto ciljnih naprav. Slednje potrjuje analiza spletne aplikacije GitHub 2.0 (Zapponi, 2020), ki kot najpogosteje uporabljen programski jezik na platformi GitHub navaja programski jezik JavaScript,



Slika 1: Graf rasti novih uporabnikov na platformi GitHub v letih 2014–2018 (GitHub, 2020a)

ki mu sledita Python in Java. Razvrstitev z deležem in njegovo spremembo glede na prejšnje četrletje desetih najpogostejših programskih jezikov platforme GitHub za obdobje prvega četrletja 2020 prikazuje Tabela 1. Heterogenost projektov sicer pripomore k lažjemu prilagajanju razvojnih orodij različnim okoljem in razvojnim procesom, v okviru katerih se ta uporabljajo.

Tabela 1: Deset najpogostejših programskih jezikov na platformi GitHub za obdobje prvega četrletja 2020 z navedenimi deleži programskih jezikov in spremembo deleža glede na prejšnje četrletje

mesto uvrstitve	programski jezik	delež	sprememba
1	JavaScript	18,7 %	-1,41
2	Python	16,2 %	-1,65
3	Java	10,9 %	+0,53
4	Go	9,0 %	+0,98
5	C++	7,4 %	+0,04
6	Ruby	6,8 %	+0,34
7	TypeScript	6,7 %	+1,52
8	PHP	5,1 %	-0,46
9	C#	3,8 %	+1,14
10	C	3,2 %	-0,20

2.3 Samoučeča razvojna orodja

Z uporabo strojnega učenja in podatkov, pridobljenih z rudarjenjem razpoložljivih baz znanja, ki jih soustvarjamo v sklopu omenjenih spletnih skupnosti in repozitorijev, je mogoče zgraditi razvojna orodja, ki predstavljajo močan odmik od trenutno uveljavljenih razvojnih orodij. Ta namreč temeljijo na vnaprej kodiranih pravilih obnašanja in strategijah reševanja problemov. Za zagotavljanje skladnosti obnašanja razvojnih orodij s pričakovani uporabnikov skrbi jo razvijalci razvojnih orodij, ki morajo vsako nadgradnjo ali spremembo obnašanja programski kodi popraviti ročno. Razvojna orodja, ki temeljijo na pristopu strojnega učenja se temu izognejo. Izpeljava potrebnih pravil, ki pripeljejo do rešitve zastavljenega cilja, je v celoti prepuščena procesu strojnega učenja, ki je pri tem avtonomen. Od orodij za razvoj programske opreme naslednje generacije si obetamo odpravo determinističnega in vnaprej programiranega obnašanja. Od tovrstnih orodij pričakujemo zmožnost samodejnega učenja reševanja razvojnih izzivov in samodejno prilagajanje okoliščinam dela.

3 PODPORA UMETNE INTELIGENCE RAZVOJNEMU CIKLU PROGRAMSKE OPREME

Razvoj programske opreme velja za kompleksen proces, uspešnost katerega je odvisna od različnih faktorjev, pomembnejši med njimi so zagotovo razvojna skupina, vrsta programskega produkta v razvoju, stabilnosti funkcionalnostnih zahtev, izbira programskega jezika in arhitekture programske rešitve (Güemes-Peña idr., 2018). Realizacija projektov programskih produktov torej terja skupino dobro izurjenih strokovnjakov iz področja programskega inženirstva, od katerih se ob dobro razvitih komunikacijskih sposobnostih in obvladovanju dela v skupini pričakujejo predvsem močne razvite sposobnosti analitičnega razmišljanja ter zmožnost reševanja problemov, na katere naletijo med razvojem programskih produktov. Potreba po navedenih veččinah izhaja iz narave dela, ki terja od razvijalcev programske opreme v prvi vrsti vložen miselni napor. In ravno razbremenitev stopnje miselnega navora, ki ga morajo razvijalci vložiti v svoje delo, ostaja osrednja točka razvoja naprednih z umetno inteligenco podprtih razvojnih orodij.

Zaradi miselno intenzivne narave dela tekom razvoja programske opreme so možnosti uporabe orodij dokaj omejene, običajno na trivialna in pogosto ponavljajoča se opravila. Večino miselno zahtevnih opravil tako še vedno opravijo razvijalci. Čez leta smo bili sicer priča postopni evoluciji pristopov in orodij, sami temelji le-teh pa se čez leta niso kaj dosti spreminjali. Posledično lahko opazimo, da se pristopi programskega inženirstva in orodja, ki se pri tem uporabljajo, od samih začetkov programskega inženirstva, torej vse od vpeljave programskih jezikov Fortran in Lisp sredi petdesetih let prejšnjega stoletja, konceptualno niso korenito spremenili (Lorica & Loukides, 2018). Z večjimi ali manjšimi variacijami razvijalci povečini uporabljajo urejevalnike, v katerih urejajo izvorno kodo programov.

3.1 Podpora orodij razvoju programskih rešitev

Ker osrednji del celotnega procesa razvoja programskih rešitev še vedno predstavlja pisanje in urejanje programske kode, je to točka razvojnega procesa programske opreme, ki ponuja največ možnosti za povečanje učinkovitosti razvoja. Skrajšanje časa, v katerem razvijalci pridejo do potrebne programske kode, predstavlja enega izmed ključnih prispevkov k optimizaciji razvojnega procesa. Ključni izziv, ki jih

na tem mestu rešujejo z umetno inteligenco podprta orodja, je, kako čim hitreje do relevantne programske kode, ki je prilagojena kontekstu zahtev uporabnikov in potreb naročnikov (Gradišnik, Karakatič, idr., 2019).

3.1.1 Inteligentni asistenti

Pomembno skupino z umetno inteligenco podprtih razvojnih orodij, ki so iz idejnih zasnov že prešle v konkretne rešitve, predstavljajo inteligentni asistenti razvijalca. Njihova primarna naloga je, da v realnem času glede na kontekst programskega problema razvijalcu nudijo relevantne predloge blokov programske kode, priporočila in dobre prakse (Yao, 2018). Tovrstna razvojna orodja presegajo že uveljavljene rešitve predlagalnikov sledečega žetona programske kode. Njihovi predlogi zajemajo celostno rešitev v obliki sintaktično pravilnega bloka programske kode, ki ga je v danem kontekstu mogoče uporabiti.

Inteligentni asistenti so v pomoč predvsem, ko se razvijalci znajdejo na obsežnih in kompleksnih projektih, na katerih se morajo za nadgradnjo ali popravke posameznih funkcionalnosti prebiti čez številne vrstice programske kode. Te po možnosti niti niso napisane v programskem jeziku, ki bi jim bil povsem domač. Brskanje za rešitvami razvojnih problemov, dostopnih v najrazličnejših virih, je torej del dnevne rutine razvijalcev (Nguyen, Di Rocco, & Di Ruscio, 2018) software developers frequently look up external sources for related information. Consulting data available at open source software (OSS). Inteligentni

asistenti torej razvijalcem v takšnih situacijah nudijo asistenco na mestu, ko bi ta moral prekiniti programiranje z namenom, da v dokumentaciji ali katerem izmed spletnih virov poišče rešitev problema, ki bi mu omogočila nadaljevanje dela. Ključna naloga inteligentnega asistenta je torej ponuditi izsek programske kode, ki razvijalcu omogoči takojšnje nadaljevanje dela (Gradišnik, Tina, & Karakatič, 2019). Ponujena rešitev temelji na analizi podobnih blokov programske kode učnih projektov, pridobljenih iz analiziranih repozitorijev projektov programske opreme.

Pristop razvoja programske kode, pri katerem razvijalcu z relevantnimi predlogi stoji ob strani inteligentni asistent, bi lahko primerjali s predvsem v agilnih metodah razvoja dobro uveljavljenim principom programiranja v paru. Pri razvoju programske kode nad skupnim programskim problemom delata dva razvijalca, od katerih prevzame eden aktivno vlogo pisanja programske kode, drugi razvijalcev v paru pa spremlja delo prvega ter z aktivnim spremljanjem in predlogi skrbi za učinkovitejši in bolj kakovosten potek dela. Lahko bi rekli, da s tem ohranimo učinkovitost in raven kakovosti razvite kode, hkrati pa se izognili neekonomičnosti programiranja v paru z dvema razvijalcem, kar je eden ključnih pomislov pri njegovi vpeljavi v praksi (Handy, 2018). Slika 2 prikazuje primer predlogov inteligentnega asistenta Codota. Ta razvijalcu predlaga pomensko pravilne in popolne bloke programske kode, ki povežejo dva v kodi s strani razvijalca uporabljena razreda za delo s tokovi.

```
import java.io.File;
import java.util.zip.ZipInputStream;

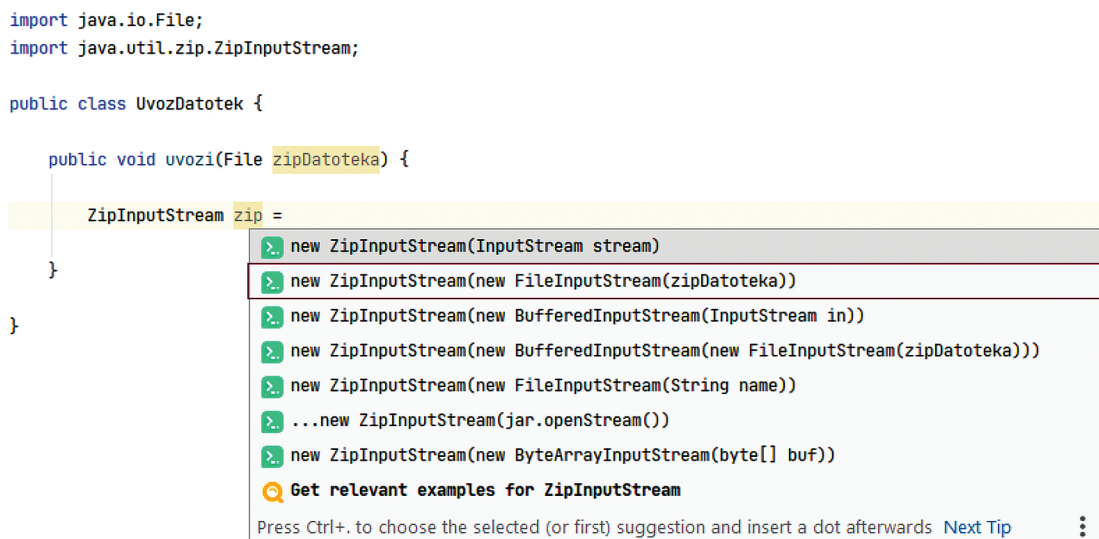
public class UvozDatotek {

    public void uvozi(File zipDatoteka) {

        ZipInputStream zip =

    }

}
```



Slika 2: Predlogi blokov kode pametnega asistenta Codota nameščenem v razvojnem okolju IntelliJ IDEA

Trenutno ima namreč vsak izmed razpoložljivih inteligentnih asistentov platformo, v kateri je dominanten. Inteligentni asistent Codota («Codota», 2019) nudi asistenco razvijalcem, ki razvijajo rešitve s pomočjo programskega jezika Java in njemu sorodnem jeziku Kotlinu. Ob podpori programskega jezika Java se lahko uporabniki inteligentnega asistenta Codota nadejajo podpore programskega jezika JavaScript, kateremu podpora se nahaja v beta fazi razvoja in bo uporabnikom na voljo kmalu. Za razvijalce, ki prisegajo na tehnološke rešitve iz Microsoftovega tabora, bo najprimernejši inteligentni asistent IntelliCode (Microsoft, 2019). Microsoftovo rešitev IntelliCode odlikuje široka podpora programskim jezikom s poudarkom na programskem jeziku C# ter odlična integracija v Microsoftovi integrirani razvojni okolji Visual Studio in Visual Studio Code. Kite (Kite, 2019), kot tretji izmed vidnejših predstavnikov inteligentnih asistentov, cilja na iz leta v leto številčnejšo populacijo Python razvijalcev (Tiobe, 2020). Kljub trenutni specializiranosti inteligentnih asistentov v posamezne programske jezike se v prihodnje kaže jasen trend, da inteligentni asistenti presežejo specializacijo v posamezne programske jezike in postanejo bolj univerzalni s tem, da ponudijo podporo širši paleti programskih jezikov. Strnjena primerjava navedenih inteligentnih asistentov prikazuje Tabela 2.

Vse navedene inteligentne asistente je možno namestiti kot vtičnike integriranih razvojnih okolij za platforme programskih jezikov, na katere ti ciljajo. S tem inteligentni asistenti dobijo vpogled v delo razvijalca in dosežejo, da so razvijalcu tekom razvoja programske kode res ves čas razvoja na voljo. V trenutni fazi zrelosti inteligentnih asistentov si lahko razvijalci od le-teh obetajo predloge blokov program-

ske kode, ki bi jih lahko v neki točki pisanja programske kode uporabili. Predlogi, ki jih dajejo inteligentni asistenti, presejajo enostavno dopolnjevanje kode. Glede na kontekst programskega problema so predlogi inteligentnih asistentov kompleksnejše strukture programske kode, do katerih so se predstavljeni inteligentni asistenti dokopali s strojnimi učenjem bodisi nad programsko kodo odprtokodnih projektov repozitorijev projektov programske opreme GitHub in Bitbucket, bodisi kot v primeru inteligentnega asistenta Codota nad programsko kodo objavljeno v razpravah med razvijalci v spletni skupnosti StackOverflow.

3.1.2 Semantični iskalniki po programski kodi

Dopolnjevanje in izpopolnjevanje obstoječih sistemov terja veliko iskanja po programski kodi, pri čemer je popolnoma običajno, da večji programski produkti obsegajo več kot 100.000 vrstic programske kode. Pri tem, ko razvijalci razmišljajo v naravnem jeziku, je programska koda, po kateri iščejo, napisana v programskem jeziku. Razvijalec mora torej v iskalnik vpisati konkretni izsek iskalne programske kode, kar pa zna biti problematično, če se z iskano programsko kodo predhodno še ni srečal. Razkorak med naravnim in programskim jezikom pri iskanju po programski kodi premošča rešitev Microsoftove podružnice GitHub poimenovan Semantic Search (Husain & Wo, 2018). Rešitev stavke programske kode pretvori s pomočjo vektorjev preslikav v naravni jezik. Na podlagi primerjave sorodnosti vektorjev programske kode in iskalnih nizov je nato mogoče najti tisto programsko kodo hranjeno v javnih repozitorijih GitHub portala, ki najbolj ustreza vnesenemu iskalnemu nizu.

Tabela 2: Strnjena primerjava inteligentnih asistentov

Inteligentni asistent	Podprti programski jeziki	Podpora v prihodnje	Podprta razvojna orodja	Viri strojnega učenja
Codota	Java Kotlin	JavaScript <i>(beta)</i>	Intellij IDEA, Android Studio, Eclipse IDE	GitHub, Bitbucket, StackOverflow, Iskalnik Google
Kite	Python	<i>V fazi izbire podpore naslednjega programskega jezika</i>	IntelliJ, PyCharm, VS Code, Atom, Vim, Sublime, Spyder	Javni repozitoriji na GitHub-u
Microsoft IntelliCode	C#, C++, Python, JavaScript, TypeScript, XAML, Java.	/	Visual Studio 2019 (C#, C++, JavaScript, XAML), Visual Studio Code (Java, Python)	GitHub (za C# več kot 3000 najboljših odprtokodnih repozitorijev), sicer na zahtevo uporabnikov tudi privatni repozitoriji uporabnika

3.1.3 Generiranje programske kode na podlagi skic uporabniškega vmesnika

Razvijalci pogosto na podlagi predhodno pripravljenih skic uporabniškega vmesnika, ki jih pripravijo grafični oblikovalci v procesu zajema zahtev stranke, sprogramirajo ustrezno izvršljivo programsko kodo. Prihranek pri porabljenem času za razvoj obljublja orodje Sketch2Code (»Sketch2Code«, 2019), ki je zmožno iz ročno skicirane risbe uporabniškega vmesnika generirati ustrezno HTML programsko kodo vključno s pripadajočimi kaskadnimi stilskimi predlogami (CSS). Ključna prednost orodja je, da ne potrebuje modela ali natančno izrisane slike prototipa, temveč zadostuje le njegova skica (Kumar, 2018). Primer pretvorbe prikazuje Slika 3. Tako generirana programska koda lahko razvijalce pripelje bodisi do hitrega prototipa rešitve bodisi je uporabljena kot skelet programske kode končne rešitve. Opisan pristop generiranja uporabniškega vmesnika so nadgradili s pomočjo tri-slojnega sistema umetne inteligence (Moran, Bernal-C'Ardenas, Curcio, Bonett, & Poshyvanyk, 2018). Pri tem za učenje orodja seveda niso uporabili znanja ekspertov, temveč javno dostopne repozitorije programske kode.

4 IZBIRA USTREZNEGA RAZVOJNEGA ORODJA

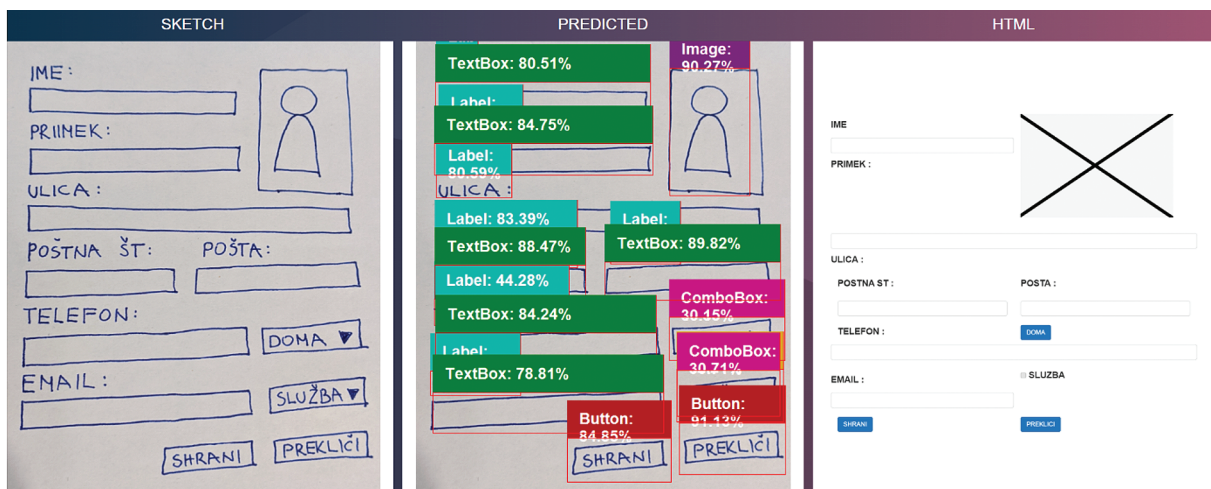
4.1 Izbira orodij

Čeprav se na prvi pogled zdi, da je trenutno na trgu več alternativnih inteligentnih asistentov, podrobnejši pregled razkrije, da so razvijalci pri izboru razpoložljivih razvojnih orodij relativno omejeni. Predsta-

vljena orodja v tem trenutku še ne nudijo prav visoke stopnje univerzalnosti in so bolj ali manj vezana na ozko usmerjena opravila, ki so dodatno vezana na izbrane programske jezike oz. programske platforme.

Kot primer vzemimo orodje Sketch2Code, ki pretvarja skice uporabniškega vmesnika v izvršljivo programsko kodo, zapisano v obliki označevalnega jezika HTML. Čeprav je lahko vhod v orodje poljubna ročno narisana skica uporabniškega vmesnika, je izhod orodja ozko usmerjen v razvijalce spletnih rešitev. Razvijalci namiznih ali mobilnih aplikacij si z orodjem ne bi mogli veliko pomagati, poiskati bi morali katero izmed alternativ.

Podobno omejeni so uporabniki inteligentnih asistentov. Podroben pogled na področje razkrije, da ključni igralci na področju naslavljajo določene skupine razvijalcev. In sicer, rešitev Codota v prvi vrsti naslavlja razvijalce programskih jezikov Java in sorodni Kotlin, s katerimi ti razvijajo zelo širok spekter javanskih in android aplikacij. Rešitev Kite primarno naslavlja iz leta v leto večjo skupnost Python razvijalcev, medtem ko IntelliCode naslavlja predvsem razvijalce, ki svoje rešitve temeljijo na Microsoftovem tehnološkem skladi in razvojnem okolju Visual Studio. Kljub začetni ozki usmerjenosti inteligentnih asistentov na ciljno skupino razvijalcev se za prihodnost kažejo njihove močne tendence, da bi postali univerzalni in primernejši za širši krog razvijalcev. Trenutno razpoložljivi inteligentni asistenti se torej primarno osredotočajo na programske jezike, razvijalci pa ostajajo zaprti znotraj okvirjev razpoložljivih rešitev, ki so za njihovo razvojno platformo na voljo.



Slika 3: SketchCode iz skice grafičnega vmesnika prepozna elemente vmesnika in na podlagi prepoznave generira HTML kodo

Pri tem je popolnoma vseeno, za kakšen tip aplikacij jih bodo razvijalci uporabili. Iz vidika uporabe inteligentnih asistentov je sicer popolnoma vseeno, ali razvijalec razvija namizno, spletno ali mobilno aplikacijo, dokler pri tem uporablja s strani inteligentnega asistenta podprt programski jezik.

4.2 Omejitve z umetno inteligenco podprtih orodij

V raziskovalni skupnosti na področju umetne inteligence že desetletja sledijo viziji, da bi s pomočjo umetne inteligence izgradili sisteme, ki bi bili zmožni avtonomno spisati računalniške programe (Balog, Gaunt, Brockschmidt, Nowozin, & Tarlow, 2019). Čeprav se v povezavi s prehodno predstavljenimi orodji pogosto uporablja besedna zveza »umetna inteligenca«, praktični preizkus razpoložljivih orodij pokaže, da so njihove zmožnosti še zmeraj zelo omejene.

Inteligentni asistenti na primer pogosto ponudijo več alternativnih rešitev. Razvijalcem so torej ponujene možne alternative rešitve in ne optimalne rešitve programskega problema v konkretnem kontekstu programskega produkta. Razumevanje programske kode se med razvijalcem in inteligentnim asistentom pomembno razlikuje. Inteligentni asistenti so v trenutni fazi razvoja zmožni razumeti zgolj ozki okvir pisanja programske kode v dometu nekaj vrstic predhodno napisane programske kode. Posledično ponudi tiste bloke programske kode, za katere obstaja glede na projekte iz učnih primerov statistično gledano največja verjetnost, da se bodo v nadaljevanju uporabili v programski kodi. Razvijalec mora na drugi strani, da je sploh sposoben realizirati programsko rešitev v skladu s pričakovanji naročnika, razumeti popolni kontekst programskega produkta, vključno z namenom uporabe programske rešitve, pričakovanim obnašanjem funkcionalnosti in željami in pričakovanji uporabnikov. Za razliko od inteligentnih asistentov razvijalec torej opremljen z razumevanjem celotnega konteksta razvoja programske rešitve razume končni cilj procesa, zaradi česar se je med možnimi alternativnimi zmožen odločiti za tiste rešitve, ki so za dani kontekst najbolj optimalne.

Kljub temu predstavlja sinteza znanja iz razpoložljivih baz znanja in priprava alternativ rešitev, ki jo opravijo inteligentni asistenti, velik prispevek k optimizaciji razvoja programske kode in je ne gre kar zanemariti. Že en sam klic programskega vmesnika, ki razvijalcu ni popolnoma poznan, pomeni prekinitve pisanja kode in posledično njegovo brskanje po

virih. Z uporabo inteligentnih asistentov je takšnega brskanja za informacijami pri programskega jezika ne popolnoma večjih razvijalcih občutno manj, vnaprej pripravljeni bloki kode pa tudi prihranijo marsikateri pritisk na tipkovnico.

5 SKLEP

Razvoj spletne tehnologij v preteklih desetletjih na čelu s spletnimi forumi, blogi in drugimi spletnimi skupnostmi je pospešil pretok znanja med razvijalci. Od naslednje generacije z umetno inteligenco podprtih orodij se pričakuje, da bodo razvijalcu v tej poplavi informacij, ki so jih spletne skupnosti akumulirale tekom več desetletij, pomagale v skladu s kontekstom problema narediti učinkovito agregacijo informacij, pridobljenih iz najrazličnejših virov v konkretno rešitev programskega problema.

Hiter razvoj algoritmov umetne inteligence in pristopov učinkovitega rudarjenja repozitorijev programske rešitve odpirata čedalje več možnosti vplejave tovrstnih tehnologij v področje programskega inženirstva. Želja po hitrejšem in cenejšem razvoju programske rešitve, višja kakovost programskega produkta in pomanjkanje razvijalcev s potrebnim naborom veščin so samo nekateri izmed izzivov, za katere si obetamo, da jih bomo lahko z vpeljavo umetne inteligence uspešneje obvladovali. Prav tako kot v drugih inženirskih vejah, so poleg ustrezno usposobljenih razvijalcev tudi v programskem inženirstvu za uspešnost projektov ključna učinkovita orodja. Ker je razvoj informacijskih rešitev področje, ki od razvijalcev terja predvsem vložek intelektualnega dela, si veliko obetamo prav od vpeljave umetne inteligence v razvojna orodja. Orodja, podprta z metodami umetne inteligence so namreč eden izmed načinov, ki učinkovito izboljšata učinkovitost razvijalce. Na tehnologijah in pristopih umetne inteligence je mogoče zasnovati novo generacijo razvojnih orodij, ki v proces razvoja informacijskih rešitev vpeljejo višjo stopnjo avtonomnosti in napredno avtomatizacijo opravil.

Vpeljava umetne inteligence v orodja za razvoj programskega produkta na tej stopnji razvoja ne bo razrešila problema pomanjkanja kakovostnih in dobro izurjenih IT strokovnjakov v splošnem. Ti ostajajo osrednji in nepogrešljiv člen v procesu razvoja programske opreme, na katerih ostaja ključne odločitve v procesu. Nepoznavanja programskega jezika ali principov razvoja programske opreme s tovrstnimi

orodji ni moč nadomestiti. Vloge sodobnih z umetno inteligenco podprtih razvojnih orodij je, da razvijalce pri njihovem delu učinkovito podprejo in jih čim bolj razbremenijo. Razvijalci lahko tako z njihovo pomočjo v krajšem času ustvarijo več. Da je ta podpora tovrstnih orodij učinkovita, postaja vedno bolj nujno, da zmorejo ponuditi rešitve ne le preprostih in ponavljajočih se opravil, temveč ponudijo rešitve za opravila, ki od razvijalca terjajo miselni napor.

LITERATURA

- [1] Balog, M., Gaunt, A. L., Brockschmidt, M., Nowozin, S., & Tarlow, D. (2019). DeepCoder: Learning to write programs. 5th International Conference on Learning Representations, ICLR 2019 - Conference Track Proceedings.
- [2] Bitbucket. (2019a). Pridobljeno 17. avgusta 2019 od <https://bitbucket.org/>
- [3] Bitbucket. (2019b). Celebrating 10 million Bitbucket Cloud registered users. Pridobljeno 2. junija 2020 od <https://bitbucket.org/blog/celebrating-10-million-bitbucket-cloud-registered-users>
- [4] Bughin, J., Hazan, E., Labaye, E., Manyika, J., Dahlström, P., Ramaswamy, S., & Cochin de Billy, C. (2016). Digital Europe: Pushing the Frontier, Capturing the Benefits. Pridobljeno 17. marca 2019 od www.mckinsey.com/mgi.
- [5] Cagle, K. (2019). Why Most Digital Transformations Will Fail. Pridobljeno 8. avgusta 2019 od Forbes website: <https://www.forbes.com/sites/cognitiveworld/2019/01/07/why-most-digital-transformations-will-fail/#4c2a2357520e>
- [6] Codota. (2019). Pridobljeno 21. januarja 2019 od <https://www.codota.com/>
- [7] Copeland, B. J. (2019). Artificial intelligence. Pridobljeno 8. avgusta 2019 od <https://www.britannica.com/technology/artificial-intelligence>
- [8] Cuesta, C., Ruesta, M., Tuesta, D., & Urbiola, P. (2015). The digital transformation of the banking industry. Pridobljeno 17. marca 2019 od <http://www.bbvaesearch.com>
- [9] Deming, D. J., & Noray, K. (2018). STEM Careers and Technological Change. Pridobljeno 17. avgusta 2019 od https://scholar.harvard.edu/files/kadeem/files/demingnoray_stem_sept2018_final.pdf
- [10] Giffi, C., Wellener, P., Dollar, B., Manolian, H. A., Monck, L., & Moutray, C. (2018). 2018 Deloitte and The Manufacturing Institute skills gap and future of work study.
- [11] GitHub. (2019). Pridobljeno 17. avgusta 2019 od <https://github.com/>
- [12] GitHub. (2020a). Thank you for 100 million repositories. Pridobljeno 1. junija 2020 od <https://github.blog/2018-11-08-100m-repos/>
- [13] GitHub, I. (2020b). GitHub is how people build software. Pridobljeno 2. junija 2020 od <https://github.com/about>
- [14] GitLab. (2019). Pridobljeno 17. avgusta 2019 od <https://about.gitlab.com/>
- [15] Gradišnik, M., Karakatič, S., Mauša, G., Beranič, T., & Heričko, M. (2019). Možnost vpeljave umetne inteligence v proces razvoja programske opreme. V Š. Urh Popovič (Ur.), Slovenija 4.0 : zbornik. 26. konferenca Dnevi slovenske informatike, 16. in 17. april 2019, Portorož. Ljubljana: Slovensko društvo Informatika.
- [16] Gradišnik, M., Tina, B., & Karakatič, S. (2019). Implementacija programskih rešitev s pomočjo inteligentnih asistentov. V M. HERIČKO & K. KOUS (Ur.), Sodobne informacijske tehnologije in storitve : OTS 2019 : zbornik štiriindvajsete konference, Maribor, 18. in 19. junij 2019 (str. 138–148). Univerzitetna založba Univerze.
- [17] Griffin, D., Mok, L., Struckman, C., & Berry, D. (2018). Tackle the Talent Problem : Invest in Growing Your Own Employees. Gartner.
- [18] Güemes-Peña, D., López-Nozal, C., Marticorena-Sánchez, R., & Maudes-Raedo, J. (2018). Emerging topics in mining software repositories: Machine learning in software repositories and datasets. *Progress in Artificial Intelligence*, 7(3), 237–247. <https://doi.org/10.1007/s13748-018-0147-7>
- [19] Handy, A. (2018). Codota Offers Pair Programming with Artificial Intelligence. Pridobljeno 14. avgusta 2019 od <https://thenewstack.io/codota-offers-ai-pair-programming/>
- [20] Husain, H., & Wo, H.-H. (2018). Towards Natural Language Semantic Code Search. Pridobljeno 18. avgusta 2019 od <https://githubengineering.com/towards-natural-language-semantic-code-search/>
- [21] Kaur, A., Kaur, K., Chopra, D., & Kaur, H. (2020). Systematic Literature Review on Mining Software Repositories. 7(1), 195–231.
- [22] Kite. (2019). Kite. Pridobljeno 17. avgusta 2019 od <https://www.kite.com/>
- [23] Kumar, A. (2018). Automated front-end development using deep learning. Pridobljeno 21. maja 2019., od <https://blog.insightdatascience.com/automated-front-end-development-using-deep-learning-3169dd086e82>
- [24] Lo Giudice, D. (2016). How AI Will Change Software Development And Applications.
- [25] Lorica, B., & Loukides, M. (2018). What machine learning means for software development. Pridobljeno 8. avgusta 2019 od <https://www.oreilly.com/ideas/what-machine-learning-means-for-software-development>
- [26] Microsoft. (2019). Visual Studio IntelliCode. Pridobljeno 18. avgusta 2019 od <https://visualstudio.microsoft.com/services/intellicode/>
- [27] Moran, K., Bernal-C'Ardenas, C., Curcio, M., Bonett, R., & Poshyvanyk, D. (2018). Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps. V IEEE TRANSACTIONS ON SOFTWARE ENGINEERING.
- [28] Nguyen, P. T., Di Rocco, J., & Di Ruscio, D. (2018). Mining software repositories to support OSS developers: A recommender systems approach. *CEUR Workshop Proceedings*, 2140.
- [29] Ross, A., & Srinivas, V. (2018). Accelerating digital transformation in banking Findings from the global consumer survey on digital banking. Pridobljeno 8. avgusta 2019 od <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-accelerating-digital-transformation-in-banking.pdf>
- [30] Sketch2Code. (2019). Pridobljeno 17. avgusta 2019 od <https://sketch2code.azurewebsites.net/>
- [31] SourceForge. (2019). Pridobljeno 17. avgusta 2019 od <https://sourceforge.net/>
- [32] StackOverflow. (2019). Pridobljeno 17. avgusta 2019 od <https://stackoverflow.com/>
- [33] Tiobe. (2020). Tiobe Index. Pridobljeno 15. marca 2020 od <https://www.tiobe.com/tiobe-index/>
- [34] Yao, M. (2018). 6 Ways AI Transforms How We Develop Software. Pridobljeno 20. marca 2019 od <https://www.forbes.com/sites/mariyayao/2018/04/18/6-ways-ai-transforms-how-we-develop-software/#21347e7d26cf>
- [35] Zapponi, C. (2020). GitHub 2.0 - A small place to discover languages in GitHub. Pridobljeno 8. junija 2020 od https://madnight.github.io/github/#/pull_requests/2019/1

■

Mitja Gradišnik je raziskovalec na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Raziskovalno se ukvarja s sodobnimi pristopi pri razvoju programskih rešitev, kakovostjo in obvladovanjem staranja programskih produktov ter praktično uporabo metod podatkovnega rudarjenja v programskem inženirstvu. Raziskovalne in aplikativno sodeluje na več projektih, ki se odvijajo v okviru Inštituta za informatiko.

■

Tina Beranič je asistentka na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Doktorirala je leta 2018 iz tematike identifikacije pomanjkljive programske kode. Njeno raziskovalno delo obsega domeno kakovosti programske opreme, še posebej področje programskih metrik in mejnih vrednosti ter njihove uporabe za namen vrednotenja programske opreme. Ukvarja se tudi s področjem revizije informacijskih sistemov, pri čemer je leta 2017 pridobila certifikat CISA (Certified Information Systems Auditor).

■

Sašo Karakatič je docent na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Raziskovalno se ukvarja s področjem umetne inteligence in strojnega učenja ter aplikacijo optimizacijskih pristopov po vzoru narave na področjih transporta in rudarjenja podatkov.

Znanjski delavci in notacija CMMN

Mateja Kocbek Bule¹

¹ Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru,
Koroška cesta 46, 2000 Maribor
mateja.kocbek@um.si

Izvleček

Primeri (angl. Cases) se uporabljajo na mnogih področjih človeškega delovanja. Najbolj nazorni so primeri iz medicine, kjer je vsak pacient obravnavan kot ločen, celovit primer. Kljub temu so splošni postopki obravnave pacientov v ozadju enaki. Strokovno osebje je tisto, ki se, na podlagi svojega znanja in izkušenj za vsak primer (vsakega pacienta) odloči, kako bo izvedla zdravljenje, ali povedano drugače, kakšna bo nova instanca tega procesa. V tem članku bomo predstavili notacijo CMMN (Case Management Model and Notation), s katero lahko modeliramo procese, ki zahtevajo več fleksibilnosti pri izvedbi. Predstavljeno bo delovanje, in tudi nekateri izzivi, ki se pojavljajo pri uporabi.

Ključne besede: znanjski delavci, menedžment primerov, Case Management Model and Notation, CMMN

Abstract

Cases are used in many areas of human operations. The most illustrative examples of cases are from medicine where every patient represents its own case. Every case requires its own operations and functions whereas humans who are involved can use their knowledge from previous cases. This article presents the standard called CMMN (Case Management Model and Notation) that covers processes where more flexibility is required. We will also discuss operations and certain challenges that we face.

Keywords: Knowledge workers, case management, Case Management Model and Notation, CMMN.

1 UVOD

Veliko področij človeškega delovanja bi lahko predstavili s primeri (ang. cases). Primer je pogost pojem, ki lahko predstavlja različne otipljive stvari ali neotipljive koncepte. Lahko opisuje neko »stvar, enoto, po kateri se lahko očitno, neposredno spozna, dokaže neka splošnejši pojem, pravilo« (Fran.si, b. d.). Npr. na spletni strani šole je objavljen primer seminarske naloge.

Nazorni primeri so iz medicine, kjer je vsak pacient s svojo boleznijo in zgodovino zdravljenja drug primer. Poznamo pa tudi primere z drugih področij, kot so pravni primeri, sodni primeri, zaposlitveni primeri, itd. (Kocbek Bule idr., 2019).

Posamezni primeri vedno vsebujejo subjekte, to so lahko ljudje, zakonske akcije, poslovne interakcije, ali druge točke preko katerih se doseže želeni cilj. Primeri zahtevajo oz. vsebujejo tudi veliko informacij. Takoj, ko strokovnjak za neko področje dobi nov primer, lahko s pomočjo znanja, ki ga je uporabil že pri predhodnem primeru, pomaga tudi pri aktual-

nem primeru. Tako se lahko primeri rešujejo hitreje in bolj učinkovito.

Našteli smo nekaj deležnikov, ki sodelujejo pri reševanju primerov. To so: informacije, akcije, človeški viri, znanje, itd. Vse lahko združimo pod poimenovanjem »menedžment primerov«, ki ga vodi skupina strokovnjakov, imenovani tudi »znanjski delavci« (ang. Knowledge workers) (Davenport, 2005) (Rant, 2002).

Ena najpomembnejših značilnosti menedžmenta primerov je načrtovanje. Vsak primer namreč zahteva določeno stopnjo fleksibilnosti, saj se primeri med seboj razlikujejo. Težko si je namreč predstavljati, da bosta prišla k zdravniku dva pacienta, ki bosta imela točno enake simptome za določeno bolezen. Fleksibilnost pri izbiri naslednjega koraka zdravljenja lahko omogoči zdravniku, da enemu pacientu določi pregled A, drugemu pa pregled B. Zaporedje dogodkov je torej pri pacientu A, drugačno kot pri pacientu B. Znanjski delavci so torej tisti, ki izberejo točno zaporedje aktivnosti procesa zdravljenja, glede na svoje znanje in izkušnje (Swenson, 2013).

Če pogledamo celotno zadevo iz vidika modeliranja procesa, ki smo ga omenili zgoraj, vidimo da proces na nizkem nivoju abstrakcije zelo težko še boljše zapišemo. Razlog je v tem, da je lahko skoraj vsaka instanca procesa drugačna oz. vsebuje morda aktivnost, ki je katera druga instanca ne vsebuje.

Na področju modeliranja poslovnih procesov je zelo poznan standard BPMN (ang. Business Process Model and Notation) (OMG (Object Management Group), 2011), ki omogoča modeliranje poslovnih procesov. Standard zahteva zelo dobro poznavanje procesa, da lahko kasneje iz nekega nestrukturiranega besedila dobimo model. Dokler so procesi enolično določeni je uporaba takega standarda smiselna in zelo dobrodošla.

Kakor hitro pa imamo opravka s procesi, kjer sodelujejo znanjski delavci, katerih delo zahteva določeno stopnjo fleksibilnosti, pa se lahko za zapisovanje poslovnih procesov uporabi notacija CMMN (ang. Case Management Model and Notation) (Kocbek Bule idr., 2019; OMG (Object Management Group), 2016). V članku bomo tako predstavili notacijo CMMN in vlogo znanjskih delavcev v poslovnih procesih.

2 PREDSTAVITEV PODROČJA

Področje poslovnih procesov je pomembna komponenta mnogih organizacij, saj se uporaba reflektira v izdelkih, storitvah oz. končnih produktih. Življenjski cikel poslovnih procesov v nekaterih fazah uporablja *modeliranje poslovnih procesov*, z namenom boljšega nadzora nad poslovnim procesom organizacije in tudi boljšega razumevanja le-tega med zaposlenimi (Dumas idr., 2013).

Poslovne procese si pogosto še vedno predstavljamo kot zelo predvidljive, vnaprej točno določene, enake v vsaki instanci (Auer idr., 2014). To izvira iz industrijske dobe, kjer je delo običajno potekalo za tekočim trakom, kar pomeni, da je bila vsaka instanca procesa zelo podobna prejšnji. Avtor Savage kot nadaljevanje industrijske dobe navaja »dobo znanja«, ki temelji na znanju in izkušnjah strokovnjaka (Savage, 1996). V takem okolju rutinsko delo zamenja nerutinsko, oz. vsaki instanci procesa prilagojeno izvajanje.

V organizacijah, kjer je delo rutinsko in je poslovni proces naravnan predvidljivo (vsaka instanca procesa je enaka) je za modeliranje poslovnih procesov najbolj sprejet in razširjen standard BPMN (Kocbek

idr., 2015). Pri nerutinskem delu, pri procesih, kjer je zaželeno ali zahtevana fleksibilnost, pa se uveljavlja notacija CMMN.

2.1 Navezava s standardom BPMN

Glavni razlog za uvedbo notacije CMMN je bila potreba po notaciji, ki bo omogočala večjo fleksibilnost modela poslovnega procesa. Fleksibilnost je potrebna, ker se lahko poslovni proces v eni izvedbi izvede drugače, kot se bo v naslednji. Npr.: vrstni red aktivnosti je lahko drugačen, nekatere aktivnosti se izvedejo, druge ne. Ni torej vedno potrebno, da imamo strogi postopek, kjer se aktivnosti izvajajo ena za drugo.

Znanjski delavci so tisti, ki v takšnih dinamičnih/nerutinskih procesih določajo katere aktivnosti se bodo izvajale, kdaj, po kakšnem zaporedju. Nekatere aktivnosti se lahko v tudi izključi. To je glavna razlika v primerjavi s široko sprejetimi koncepti rutinskih/strukturiranih poslovnih procesov, kjer je za vsako aktivnost točno določeno kdaj, kje in zakaj se bo izvedla. Kot smo že predstavili, natančno načrtovani, strukturirani procesi niso vedno najboljša izbira pri reševanju specifičnih primerov (primeri iz medicine).

Povzamemo lahko, da je BPMN sprejet in široko uporabljen standard, ki pa kljub vsemu vsebuje nekaj pomanjkljivosti, ki jih lahko zakrpa notacija CMMN. Trenutno je le malo »kombinirane« uporabe (Methodstyle.com, 2014).

3 ZNANJSKI DELAVCI

Znanjski delavci so delavci, ki pri svojem delu uporabljajo znanje in izkušnje. Njihovo delo je kompleksnejše in ni ponavljajoče. Pomembno je, da svoje znanje nenehno izpopolnjujejo in nadgrajujejo (Rant, 2008, 2002). Znanjski delavci so neposredno povezani tudi s poslovnimi procesi svoje institucije. Proces, ki ga izvajajo, tudi zelo dobro poznajo in so tisti, ki lahko predlagajo izboljšave procesa. Iz tega sledi tudi večja odgovornost, ki jo imajo v primerjavi s klasičnimi delavci (Rant, 2008, 2002). Znanjski delavci so lastniki svojih delovnih sredstev, tj. znanje, kar jim omogoča veliko neodvisnost od organizacije v kateri delujejo.

Še nekaj dodatnih značilnosti znanjskih delavcev v primerjavi s klasičnimi, je predstavljenih v tabeli 1. Tabela je povzeta po (Rant, 2008).

Tabela 1: Kratka primerjava, klasični – znanjski delavci

Značilnost	Klasični delavec	Znanjski delavec
opravila	enostavna, ponavljajoča	kompleksna
vodenje	odrejanje in kontrola	samovodenje
status v instituciji	podrejeni	pridruženi
število predstavnikov	veliko	manj
produktivnost	zanjo skrbijo drugi	zanjo skrbijo sami
informacije	skoraj nepomembne	ključni vir za delo
komunikacija z drugimi	skoraj nepotrebna	nujna
inovacije	občasno	neprestano
odgovornost	je nimajo	odgovornost za prispevek k svojem področju
spremembe	redke	stalnica

Vir: (Rant, 2008)

4 NOTACIJA CMMN

Notacija Case Management Model and Notation (v nadaljevanju CMMN) se uporablja kot grafična predstavitev primerov (OMG (Object Management Group), 2016). Je notacija, s katero lahko predstavimo bolj dinamične poslovne procese, z dodatnimi informacijami, ki so potrebne, da se proces izvede. CMMN je bil uveden prav zaradi ad-hoc scenarijev, ki se lahko zgodijo v različnih instancah poslovnih procesov. CMMN se povezuje z znanjskimi delavci, ki za to, da se proces izvede, uporabijo veliko svojega znanja in izkušenj, saj se znajo odzvati na različne (poslovne) dogodke. Notacija CMMN jim je tukaj samo v pomoč, saj so elementi procesa definirani na takem nivoju, da lahko glede na svoje znanje vsak primer izvedejo drugače.

Konzorcij 11 podjetij je tako predstavil notacijo CMMN, ki je sedaj pod okriljem organizacije OMG. Trenutno je notacija v verziji 1.1 (december 2016) (OMG (Object Management Group), 2016).

4.1 Elementi

Nabor elementov, ki sestavljajo notacijo CMMN ni zelo obsežen, saj je notacija relativno nova in malo v uporabi. V nadaljevanju bomo na kratko predstavili elemente notacije CMMN, začenši z najpogosteje uporabljenimi. Elementi so predstavljeni tudi vizualno. Te predstavitev (slike) so ustvarjene s pomočjo orodja Signavio (Signavio Inc., b. d.).

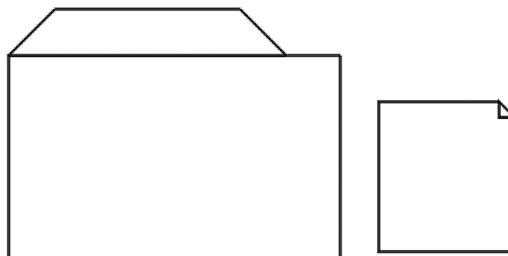
Najbolj zunanji element notacije CMMN, je *Model načrta primera* (ang. Case Plan Model) (slika 1). Vsi ostali elementi so torej uporabljani znotraj tega elementa.

Vse informacije (ali reference do njih), ki so po-

trebne za upravljanje s primerom, so definirane v *Datoteki primera* (ang. Case File). To je logični model, ki ni mišljen kot fizična shramba informacij. Vsebuje lahko več elementov *Postavka datoteke primera* (ang. Case File Items). Element je predstavljen na sliki 1. Lahko predstavlja vse: od shranjene mape ali dokumenta, shranjene celotne hierarhije sklicevanja, do vsebovanja drugih elementov tega istega tipa.

Pri modeliranju nestrukturiranih, nerutinskih procesov je pomembno, da so elementi predstavljeni tako, da je iz diagrama takoj razvidno ali so aktivnosti *zahtevane* ali *opcijske*. Elementa, ki ju je mogoče predstaviti na takšen način sta: element *Stopnja* (ang. Stage), ki ga lahko predstavimo kot eno epizodo primera in element *Opravilo* (ang. Task), ki je aktivnost znotraj primera (tabela 2).

Element *Poslušalec dogodkov* (ang. Event Listener) je element, ki se odziva na dogodke. V notaciji CMMN so trije različni takšni elementi: poslušalec navadnega dogodka, poslušalec časovnega dogodka in poslušalec uporabniškega dogodka (slika 2). S pomočjo elementa *Mejnik* (ang. Milestone) se lahko definira cilje (slika 2).



Slika 1: Element Model načrta primera in element Postavka datoteke primera

Tabela 2: **Možni prikazi elementov Stopnja in Opravilo**

Ime elementa	Navadni prikaz	Diskrecijski prikaz
Stopnja (ang. Stage)		
Opravilo (ang. Task)		

Slika 2: **Trije različni elementi Poslušalec dogodkov in element Mejnik**

CMMN prav tako vsebuje širok nabor različnih opravil (ang. Tasks), ki so predstavljena v tabeli 3. Kot je bilo predhodno že navedeno, je mogoče opravila predstaviti na dva različna načina. V tabeli 3 sta predstavljena oba načina za vsako obstoječe opravilo: prvo je *Običajno opravilo*, ki nima definiranih posebnosti; sledi *Ročno opravilo*, ki se izvede takoj po namestitvi, nad takimi opravili sistem za upravljanje s primeri nima nadzora; *Uporabniško opravilo* je opravilo, ki ga opravi uporabnik s podporo sistema za upravljanje s primeri; *Procesno opravilo* se lahko uporablja za povezovanje z modelom BPMN; *Opravilo primera* se lahko uporablja za povezovanje z drugim modelom CMMN in na koncu *Odločitveno opravilo* se lahko uporablja za povezovanje z modelom DMN.

Elementi notacije CMMN lahko vsebujejo tudi več različnih atributov (ang. decorators). Vsi so pred-

Tabela 3: **Opravila v CMMN**

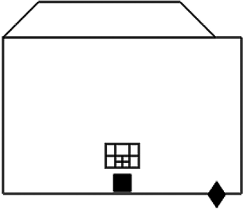
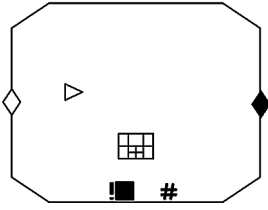
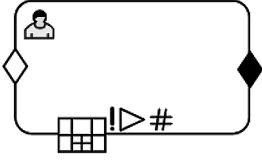

Ime in opis opravila	Navadni prikaz	Diskrecijski prikaz
Običajno opravilo (ang. Task)		
Ročno opravilo (ang. Human Task, isBlocking: false)		
Uporabniško opravilo (ang. Human Task, isBlocking: true)		
Procesno opravilo (ang. Process Task)		
Opravilo primera (ang. Case Task)		
Odločitveno opravilo (ang. Decision Task)		

stavljene v tabeli 4. V nadaljevanju, v tabeli 5, pa je predstavljena dovoljena uporaba teh atributov na elementih. Celotna vsebina obeh tabel je povzeta po viru (OMG (Object Management Group), 2016).

Tabela 4: **Obstoječi atributi**

Tabela načrtovanja	Vstopni kriterij	Izstopni kriterij	Avtomatski zaključek	Manualna aktivacija	Zahtevano	Ponavljanje
ang. Planning Table	ang. Entry Criterion	ang. Exit Criterion	ang. Auto Complete	ang. Manual Activation	ang. Required	ang. Repetition

Tabela 5: Dovoljena uporaba atributov na določenih elementih

Model načrta primera	Stopnja	Opravilo	Mejnik
ang. Case Plan Model	ang. Stage	ang. Task	ang. Milestone
			
		*Atribut Tabela planiranja se lahko uporabi le v kombinaciji z elementom Uporabniško opravilo.	

Atributa *Vstopni kriterij* (ang. Entry Criterion) in *Izstopni kriterij* (ang. Exit Criterion) spadata pod element *Strážar* (ang. Sentries), ki pazi na dogajanje, ki lahko vpliva na potek/razplet primera. Lahko bi rekli, da je *Strážar* skupek dogodka in pogoja. Atributa se lahko uporabljata le na nekaterih elementih. Dovoljena uporaba je prikazana v tabeli 5.

4.2 Delovanje

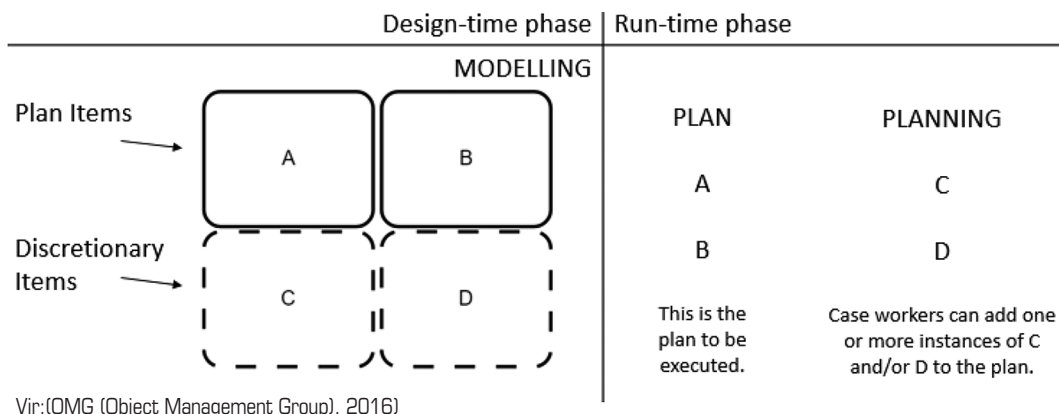
Poleg vseh predstavljenih elementov notacije CMMN so za načrtovanje in izvedbo procesov potrebna tudi pravila delovanja. Notacija CMMN pokriva procese, kjer so lahko posamezne instance procesov med seboj drugačne. Torej vsaka instance procesa se lahko izvede po drugačnem zaporedju, uporabijo se lahko druga opravila, morda se celo katero opravilo ne izvede, itd. Približek takšnega delovanja poznamo že tudi iz standarda BPMN, kjer so lahko deli procesa definirani kot »ad-hoc«. To pomeni, da ni točno določenega zaporedja izvedbe opravil. Dovoljeno je

tudi izvajanje določenega opravila večkrat. Zaporedje in frekvenco opravil, v takšnih primerih, določajo znanjski delavci (Hinkelmann, 2014).

Kot je razvidno iz slike 3 je lahko posamezen primer v dveh različnih fazah: (1) v fazi načrtovanja (ang. design-time phase), ali (2) v fazi izvedbe (ang. run-time phase) (Hinkelmann, 2014).

V fazi načrtovanja se določi, katera opravila so vedno del procesa in se morajo izvesti v vsaki instanci procesa, in katera so opcijska. Oba načina prikaza opravil sta predstavljena v tabeli 3: navadni prikaz s polno črto okoli opravila je simbol za obvezna opravila, medtem ko je diskrecijski prikaz, s črtkano črto okoli opravila, simbol za opcijsko opravilo, ki so dodatno na voljo znanjskim delavcem, glede na njihovo presojo. V fazi izvedbe se dejansko izvedejo planirana, obvezna opravila. Glede na potrebe posamezne instance procesa pa se dodatno lahko izvedejo tudi diskrecijska opravila (Hinkelmann, 2014).

Predvsem iz faze izvedbe sledi, da se v času iz-



Vir: (OMG (Object Management Group), 2016)

Slika 3: Faze primerov

vajanja procesa znanjski delavec lahko odloči, če bo vključil v trenutni proces neko opravilo ali ne. Odločitev se sprejema na podlagi znanja, ki ga je znanjski delavec pridobil s pomočjo podatkov o procesu. Referenca na podatke/informacije/znanje v procesih modeliranih z notacijo CMMN je zelo pomembna. Podatki so pogosto shranjeni v datotekah primera (ang. case file), ki se jih torej lahko uporabi pri sprejemanju odločitev znotraj procesa (Hinkelmann, 2014). Znanje za izvedbo procesa najprej predstavlja znanje, ki ga ima znanjski delavec (npr. zdravnik ima znanje medicine). Tukaj se lahko dodajajo tudi znanja ali t.i. »tiha« pravila vseh znanjskih delavcev in institucij, ki so del procesa (npr. nerazkrivanje zdravstvenega stanja pacienta) (Hinkelmann, 2014).

5 IZZIVI UPORABE CMMN ZA ZNANJSKE DELAVCE

Notacija CMMN je sorazmerno nova, kar pomeni, da je trenutno njena uporaba omejena. Prav tako je slabše poznano tudi celotno področje menedžmenta primerov. Seveda obstajajo podobnosti s standardom BPMN, vendar moramo pri modeliranju v CMMN razmišljati drugače, se bolj osredotočati na vlogo posameznika v procesu (F. A. Cummins, 2017) predictable processes. CMMN (Case Management Model and Notation.

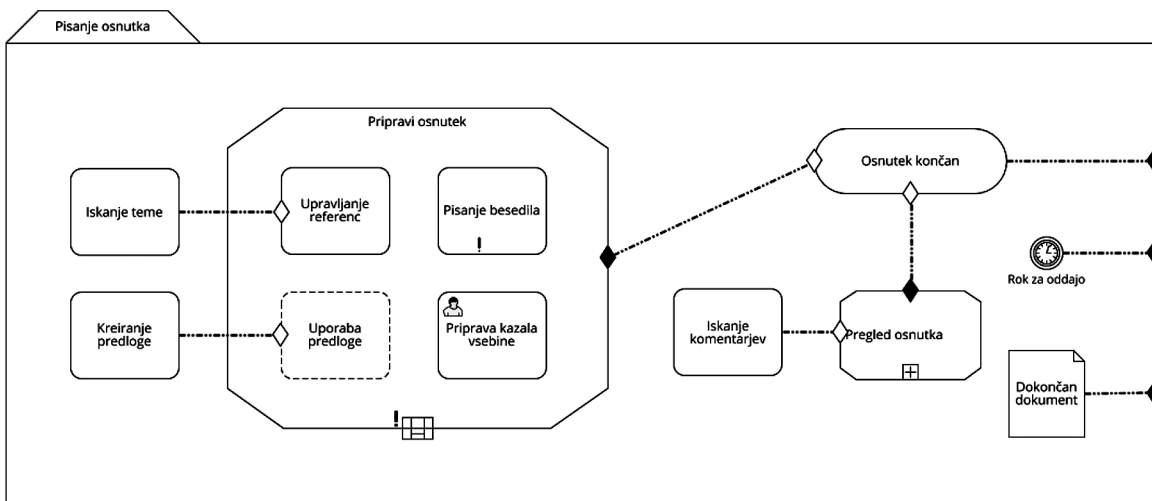
Znanjski delavci s svojim znanjem in izkušnjami prispevajo k prilagojenemu načinu izvedbe vsake instance rutinskega procesa. V rutinskih procesih so udeleženci izvajali naloge po točno določenem,

danem postopku, kjer ni bilo potrebe po vključitvi znanja in izkušenj udeležencev. To se je, kot je bilo že omenjeno, z dobo znanja, spremenilo. Interakcija je vedno bolj zaželeno oz. ponekod celo potrebna (primeri iz medicine).

Z notacijo CMMN smo dobili možnost formalnega zapisa nerutinskih procesov, ki pa se srečuje z nekaterimi izzivi (F. A. Cummins, 2017) predictable processes. CMMN (Case Management Model and Notation.

- Sodelovanje med znanjskimi delavci je slabo definirano. Znanjski delavci morajo vedeti, kdo vse sodeluje v dani instanci procesa in imeti možnost komunikacije med seboj.
- Iz zgornje alineje izhaja tudi smiselnost vključitve vlog v proces.
- Opravila bi morala vsebovati več informacij za potencialno potrebo.
- Omogočen bi moral biti vpogled v zgodovino izvedbe primera.
- Omogočena bi morala biti večja povezljivost med primeri.
- Za nekatera opravila bi bilo smiselno, da je omogočeno potrjevanje s strani drugih udeležencev ali znanjskih delavcev.
- Podobno kot pri drugih modelih je tudi model primera potrebno graditi postopno, iterativno in upoštevati odziv uporabnikov procesa.

Rešitve za navedene izzive lahko iščemo v nadgradnjah oz. razširitvah, vendar je potrebno celotno področje še dodatno raziskati in predvsem uporabljati v praksi.



Slika 4: Primer pisanja osnutka v notaciji CMMN

6 PRIMER NERUTINSKEGA PROCESA

6.1 Primer 1

K predstavitvi notacije CMMN vsekakor spada tudi primer procesa zmodeliran v CMMN. V primeru, ki je predstavljen na sliki 4, bo vključenih nekaj pogosto uporabljenih elementov. Vsi so bili predstavljenih v prejšnjih poglavjih. Primer predstavlja proces pisanja dokumenta/osnutka, ki ga vsi relativno dobro poznamo. Tako bo morda model notacije CMMN tudi lažje razumljiv. Model je ustvarjen s pomočjo orodja Signavio (Signavio Inc., b. d.).

Celoten model zaobjema element *Model načrta primera*, poimenovan 'Pisanje osnutka'. Karkoli se zgodi v tem procesu, mora ta element zajemati, ne glede na to, kaj se izvede v določenih instancah in kaj ne.

Proces se lahko začne z enim od *opravil*: 'Iskanje teme' ali 'Kreiranje predloge', izvesti pa se morata obe opravili, kar tudi vsebinsko ustreza, saj pri pisanju dokumenta potrebujemo neko temo in predlogo za pisanje.

Sledi nekoliko obsežnejši element *Stopnja*, poimenovan 'Pripravi osnutek', ki vsebuje štiri opravila in katerega izvedba je *zahtevana* (simbol klicaj). Pri pisanju dokumenta namreč potrebujemo osnutek.

Obvezno *opravilo*, poimenovano 'Upravljanje referenc', ima *vstopni kriterij* (ali tudi vstopni pogoj), ki mora biti izpolnjen, da se opravilo sploh izvede. Vsebinsko to pomeni, da moramo najprej poiskati ustrezno temo za pisanje in nato lahko upravljamo z referencami.

Podobno ima *vstopni pogoj* tudi *opravilo*, poimenovano 'Uporaba predloge', vendar to opravilo ni obvezno, kar je nakazano s črtkano črto. Opravilo se lahko v neki instanci procesa izvede, v drugi ne. Pogoj, ki mora biti izpolnjen, da je opravilo pripravljeno na izvedbo, pa je kreirana predloga.

Znotraj elementa *Stopnja* imamo tako še dve obvezni *opravili*, to sta: 'Pisanje besedila' in 'Priprava kazala vsebine'. Prvo je *zahtevano*, saj vsebuje simbol klicaj, enako kot element *Stopnja*, drugo pa je t. i. *uporabniško opravilo*, ki ga opravi uporabnik s podporo nekega sistema. Vsebinsko: obvezno moramo zapisati besedilo in pripraviti kazalo vsebine, s pomočjo izbranega orodja.

Opravilo, poimenovano 'Iskanje komentarjev' in element *Stopnja*, poimenovan 'Pregled osnutka', sta naslednja elementa. *Stopnja* ima *izstopni kriterij* (ali izstopni pogoj), ki je povezan z elementom *Mejnik*, poimenovanim 'Osnutek končan', ki pa prav tako vse-

buje *vstopni kriterij*. Vsebinsko bi lahko to pojasnili kot: zaključek pregleda osnutka vodi v končano stanje dokumenta.

Dodatno imamo na koncu še dva elementa: *Poslušalec časovnega dogodka* in *Postavka datoteke primera*. Prvi element definira datum, do katerega je potrebno dokument dokončati. Drugi pa vsebuje dejanski dokument. Oba elementa sta povezana z *izstopnim kriterijem*, kar onemogoča predhodno zaključevanje procesa. Tudi zgoraj omenjeni element *Mejnik* je v nadaljevanju povezan na izstopni pogoj iz celotnega procesa oz. iz elementa *Model načrta primera* (OMG (Object Management Group), 2016), (Marin, 2016), (Kocbek & Polančič, 2016).

6.2 Primer 2

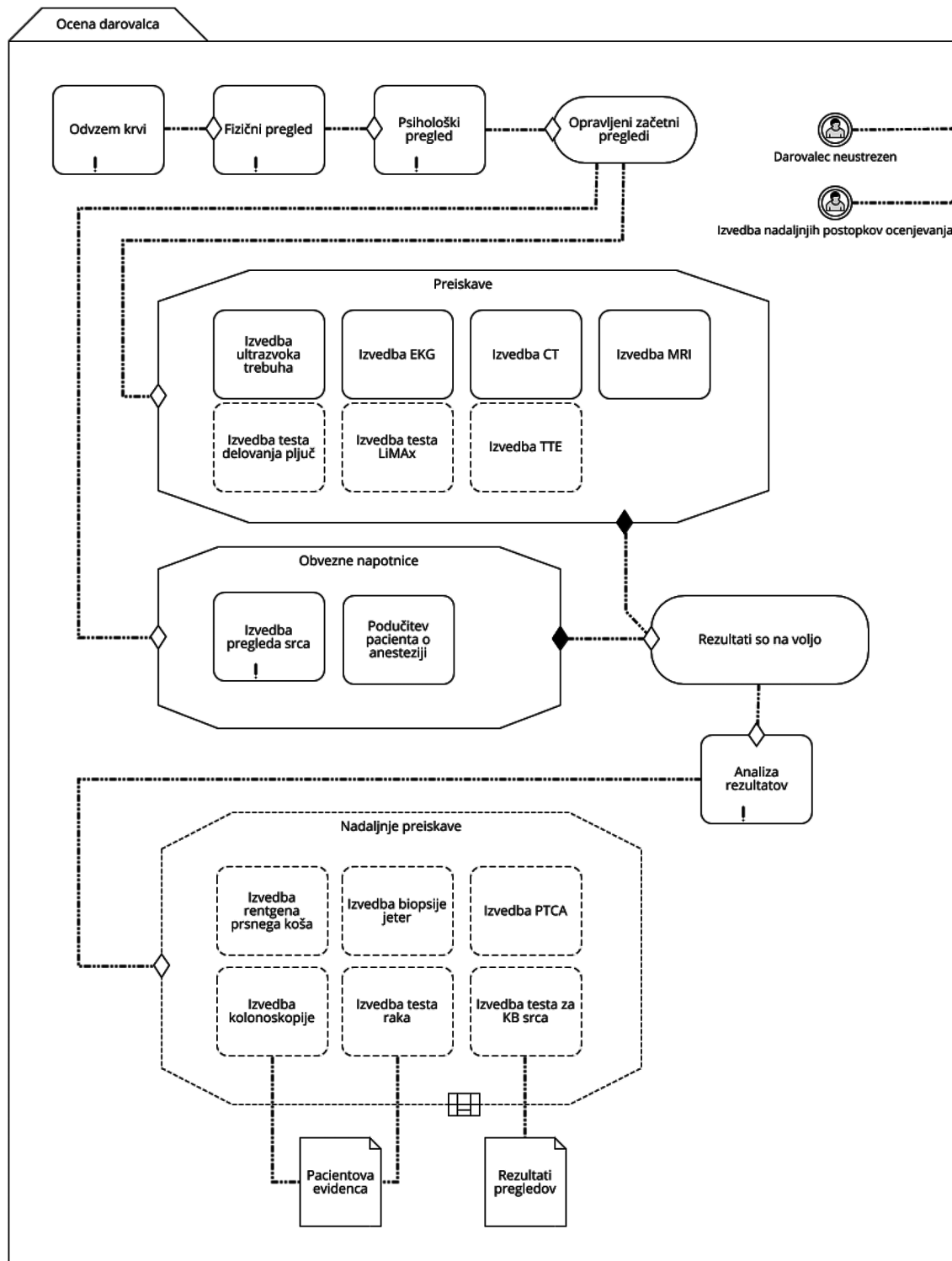
Kot drugi primer poslovnega procesa, zmodeliranega v notaciji CMMN, bi želeli predstaviti kratek proces iz zdravstva. Model predstavlja proces ocenjevanja primernosti darovalca jeter (slika 5)¹. Model je ustvarjen s pomočjo orodja Signavio (Signavio Inc., b. d.) in povzet po viru (Herzberg idr., 2014; Nešković & Kirchner, 2016)

Enako kot pri prejšnjem primeru, tudi tukaj element *Model načrta primera* zajema celoten proces. V tem primeru je nekoliko več povezav med elementi, kar nakazuje na nižjo stopnjo fleksibilnosti procesa. To je bolj izrazito v začetku procesa, kjer imamo tri povezana opravila. Vsebinsko to pomeni, da moramo opraviti nekaj začetnih pregledov, kjer je izvedba le-teh zaporedna.

Element *Mejnik*, poimenovan 'Opravljeni začetni pregledi', vodi v nadaljnja elementa *Stopnja*. Večja fleksibilnost se pokaže znotraj elementa *Stopnja*, poimenovanega 'Preiskave', kjer imamo sedem nepovezanih opravil, ki se lahko izvedejo v poljubnem vrstnem redu. Izvedba treh opravil je opcиска, kar nakazuje črtkana črta roba. Vsebinsko: da postanemo darovalec moramo na naslednje preiskave: ultrazvok trebuha, EKG, CT, MRI. Preiskave kot so test delovna pluć, test LiMax in TTE pa so opciski.

Element *Mejnik*, poimenovan 'Rezultati so na voljo', vodi v opravilo, poimenovano 'Analiza rezultatov'. Tukaj se na podlagi pridobljenih rezultatov darovalca določi ali so potrebne nadaljnje preiskave ali ne. Na elementu *Stopnja*, poimenovanemu 'Nadaljnje preiskave', se to grafično prikazuje s črtkano črto roba.

¹ Opisani proces ni primerno vodilo v dejanskih tovrstnih procesih.



Slika 5: Primer procesa v zdravstvu v notaciji CMMN

Proces se zaključi z elementom *Poslušalec uporabniškega dogodka*, poimenovanega 'Darovalec neustrezen', v kolikor je darovalec neustrezen. V nasprotnem primeru, pa se proces prav tako zaključi z

elementom *Poslušalec uporabniškega dogodka*, ki vodi v izvedbo nadaljnjih postopkov ocenjevanja darovalca oz. vodi v drug proces.

7 SKLEP

V članku smo predstavili notacijo CMMN za menedžment primerov v navezavi z znanjskimi delavci. Od rutinskih procesov, ki temeljijo na principu »tekočega traka« in so v vsaki instanci procesa enaki, smo sedaj prešli v dobo znanja, kjer so procesi vedno bolj nerutinski. Ti zahtevajo sodelovanje znanjskih delavcev, ki doprinesejo s svojim znanjem in izkušnjami. Nazorni primeri so iz medicine, kjer npr. zdravnik, ki je znanjski delavec, za vsak primer odloči, kako bo izvedel zdravljenje. Povsem razumljivo je, da bo za vsakega pacienta proces izveden nekoliko drugače.

Notacija CMMN je primerna za modeliranje tovrstnih procesov, saj omogoča več fleksibilnosti pri izvedbi procesa, kot npr. standard BPMN. Kljub temu se notacija CMMN srečuje z nekaterimi izzivi, ki smo jih predstavili v članku, saj je še relativno malo v uporabi. Izpostaviti pa želimo, da notacija CMMN lahko pomembno prispeva k spremembam na področju modeliranja poslovnih procesov, saj ponuja nekoliko drugačen pristop oz. pogled na procese, kot druge, že nekoliko bolj uveljavljene notacije.

LITERATURA

- [1] Auer, D., Hinterholzer, S., Kubovy, J., & Küng, J. (2014). Business Process Management for Knowledge Work: Considerations on Current Needs, Basic Concepts and Models. *Lecture Notes in Information Systems and Organisation*, 79–95.
- [2] Cummins, F. A. (2017). Chapter 4 - Next-Generation Business Process Management (BPM). V F. A. B. T.-B. the A. E. (Second E. Cummins (Ur.), *The MK/OMG Press* (str. 115–154). Morgan Kaufmann. <https://doi.org/https://doi.org/10.1016/B978-0-12-805160-3.00004-1>
- [3] Davenport, T. H. (2005). *Thinking for a Living: How to Get Better Performances And Results from Knowledge Workers*. Harvard Business Press.
- [4] Dumas, M., Rosa, M. La, Mendling, J., & Reijers, H. A. (2013). *Fundamentals of Business Process Management*. Springer Publishing Company, Incorporated.
- [5] Fran.si. (b. d.). *Beseda: primer*. Pridobljeno 28. februar 2020., od <https://fran.si/iskanje?View=1&Query=primer>
- [6] Herzberg, N., Kirchner, K., & Weske, M. (2014). Modeling and Monitoring Variability in Hospital Treatments: A Scenario Using CMMN. *Lecture Notes in Business Information Processing*, 3–15. https://doi.org/10.1007/978-3-319-15895-2_1
- [7] Hinkelmann, K. (2014). *Case Management Model and Notation - CMMN*. http://knut.hinkelmann.ch/lectures/bpm2013-14/06_CMMN.pdf
- [8] Kocbek Bule, M., Polančič, G., Huber, J., & Jošt, G. (2019). Semiotic clarity of Case Management Model and Notation (CMMN). *Computer Standards & Interfaces*, 66, 103354. <https://doi.org/10.1016/j.csi.2019.04.013>
- [9] Kocbek, M., Jošt, G., Heričko, M., & Polančič, G. (2015). Business process model and notation: The current state of affairs. *Computer Science and Information Systems*, 12(2), 509–539.
- [10] Kocbek, M., & Polančič, G. (2016). Introduction to Case Management Model and Notation. *Information Society 2016*.
- [11] Marin, M. A. (2016). *Introduction to the Case Management Model and Notation (CMMN)*. 1–23. <http://arxiv.org/abs/1608.05011>
- [12] Methodandstyle.com. (2014). *BPMN and CMMN Compared*. <http://brsilver.com/bpmn-cmmn-compared/>
- [13] Nešković, S., & Kirchner, K. (2016). Using context information and CMMN to model knowledge-intensive business processes. *Proceedings of the 6th International Conference on Information Society and Technology ICIST, August*, 17–21.
- [14] OMG (Object Management Group). (2011). *Business Process Model and Notation Specification*. <http://www.omg.org/spec/BPMN/2.0>
- [15] OMG (Object Management Group). (2016). *Case Management Model and Notation 1.1*. 126. <http://www.omg.org/spec/CMMN/>
- [16] Rant, Ž. (2008). Prenos znanja kot dilema znanjskih delavcev in učeče se organizacije. *Organizacija*, 41(2), str. A126-A131. <http://organizacija.fov.uni-mb.si/index.php/organizacija-si/article/viewFile/753/654>
- [17] Rant, Ž. (2002). Znanjski delavci. *Management in Evropska unija : zbornik konference z mednarodno udeležbo / 21. znanstvena konferenca o razvoju organizacijskih ved*, 409–418.
- [18] Savage, C. M. (1996). *Fifth Generation Management: Co-creating Through Virtual Enterprising, Dynamic Teaming, and Knowledge Networking*. Butterworth-Heinemann.
- [19] Signavio Inc. (b. d.). *Academic Signavio*. academic.signavio.com
- [20] Swenson, K. D. (2013). State of the Art In Case Management. V *Fujitsu America, Inc* (Številka March). <http://www.fujitsu.com/downloads/INTSTG/bop/Fujitsu-Interstage-State-of-the-Art-In-Case-Management.pdf>

Mateja Kocbek Bule je asistentka in doktorska študentka na Fakulteti za elektrotehniko, računalništvo in informatiko, Univerze v Mariboru. Med njene interesne dejavnosti spada predvsem področje upravljanja poslovnih procesov.

Application of tropical semiring for matrix factorization

Amra Omanović, Polona Oblak, Tomaž Curk

University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia
 amra.omanovic@fri.uni-lj.si, polona.oblak@fri.uni-lj.si, tomaz.curk@fri.uni-lj.si

Abstract

Matrix factorization methods employ standard linear algebra, i.e. linear models, for recommender systems. With the introduction of the tropical semiring, we can achieve non-linearity. We review algorithms that use the tropical semiring for matrix factorization and provide their strengths and limitations. We show that the tropical matrix factorization yields better results than non-negative matrix factorization for the synthetic data created by the underlying process of the tropical semiring.

Keywords: Data embedding, data mining, matrix factorization, subtropical semiring, tropical semiring.

Izvleček

Metode matrične faktorizacije uporabljajo za priporočilne sisteme standardno linearno algebro, torej linearne modele. Z zamenjavo operacij in z uvedbo tropskega polkolobarja lahko dodamo metodam komponento nelinearnosti. V članku pregledamo algoritme, ki za faktorizacijo matrike uporabljajo tropski polkolobar, in podamo njihove prednosti in omejitve. Pokažemo, da tropska matrična faktorizacija daje boljše rezultate kot nenegativna matrična faktorizacija na sintetičnih podatkih, ustvarjenih z množenjem matrik v tropskem polkolobarju.

Ključne besede: Vložitev podatkov, podatkovno rudarjenje, matrična faktorizacija, subtropski polkolobar, tropski polkolobar

1 INTRODUCTION

Data mining is one of the main challenges in computer science. There is a need to develop methods to embed data into a lower-dimensional latent space, which may help with various machine learning tasks. A data embedding model, such as matrix factorization (MF), gives us a more compact representation of the data and simultaneously finds a latent structure. MF algorithms (*e.g.*, non-negative matrix factorization (NMF) [Lee and Seung, 1999]) decompose the original matrix into a product of a base matrix and a coefficient matrix of lower dimensions. Most of machine learning methods for data embedding, *e.g.*, [Lee and Seung, 1999, Žitnik and Zupan, 2015, Zhang et al., 2007, Laurberg et al., 2008], use standard linear algebra.

Recently, several authors considered substituting the standard linear algebra with other semiring operations, *e.g.*, [Karaev and Miettinen, 2016a, Karaev et al., 2018, Karaev and Miettinen, 2016b, Karaev and Miettinen, 2019]. In this paper we review some algo-

rithms that use alternative nonstandard operations for matrix factorization and provide their strengths, limitations and potential of discovering interesting patterns. In our work, we are motivated by the question from [Karaev and Miettinen, 2016a], asking if a tropical matrix factorization can be used except for data analysis, also in other data mining and machine learning tasks, such as matrix completion. We expect that for the data that is not normally distributed and may contain a lot of extreme values using tropical semiring should give better results than MF methods that use standard operations of addition and multiplication.

Standard MF methods belong to the class of linear models that are unable to model complex relations. With the tropical semiring, we can introduce the non-linearity using the maximum operator. Another motivation for using tropical semiring is the work of Zhang *et al.* [Zhang et al., 2018]. They showed that linear regions of feedforward neural networks with rectified linear unit activation correspond to vertices

of polytopes associated with tropical rational functions. Therefore, to understand specific neural networks, we need to understand relevant tropical geometry. Since the goal is not just to model the data, but also to understand the underlying mechanisms, the matrix factorization methods that use tropical semiring can give us a more straightforward interpretation than neural networks.

We split the remainder of the paper into the following sections. Sections 2 and 3 describe tropical semiring and related work, followed by results in Section 4. We conclude our paper in Section 5.

2 TROPICAL SEMIRING

The $(\max, +)$ semiring or tropical semiring \mathbb{R}_{\max} is the set $\mathbb{R} \cup \{-\infty\}$, equipped with \max as addition (\oplus), and $+$ as multiplication (\otimes). For example, $2 \oplus 3 = 3$ and $1 \otimes 1 = 2$. On the other hand, in the subtropical semiring or (\max, \times) semiring, defined on the same set $\mathbb{R} \cup \{-\infty\}$, addition (\max) is defined as in the tropical semiring, but the multiplication is the standard multiplication (\times). By taking the logarithm of the subtropical semiring, we obtain the tropical semiring, thus these two semirings are isomorphic.

Let $\mathbb{R}_{\max}^{m \times n}$ define the set of all $m \times n$ matrices over tropical semiring. For $A \in \mathbb{R}_{\max}^{m \times n}$ $R^{m \times n}$ we denote by a_{ij} the entry in the i -th row and the j -th column of matrix A . We define the sum of matrices $A = [a_{ij}]$, $B = [b_{ij}] \in \mathbb{R}_{\max}^{m \times n}$ as

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max \{a_{ij}, b_{ij}\},$$

$i = 1, \dots, m, j = 1, \dots, n$, and the product of matrices

$$A = [a_{ij}] \in \mathbb{R}_{\max}^{m \times p}, B = [b_{kl}] \in \mathbb{R}_{\max}^{p \times n} \text{ as}$$

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^p a_{ik} \otimes b_{kj} = \max_{1 \leq k \leq p} \{a_{ik} + b_{kj}\},$$

$i = 1, \dots, m, j = 1, \dots, n$.

Matrix factorization over a tropical semiring is a decomposition of a form $A = U \otimes V$, where $A \in \mathbb{R}_{\max}^{m \times n}$, $U \in \mathbb{R}_{\max}^{m \times r}$, $V \in \mathbb{R}_{\max}^{r \times n}$ and $r \in \mathbb{N}_0$. For small values of r such decomposition may not exist. A problem of tropical matrix factorization is thus stated as follows: given a matrix $A \in \mathbb{R}_{\max}^{m \times n}$ and $r \in \mathbb{N}_0$ find $U \in \mathbb{R}_{\max}^{m \times r}$ and $V \in \mathbb{R}_{\max}^{r \times n}$ such that

$$A \cong U \otimes V. \quad (1)$$

Similarly, we define a subtropical matrix factorization. Note that the factorization in tropical semiring give different results and works with different

methods than the factorization in subtropical semiring.

3 RELATED WORK

The most common examples of matrix factorization are the singular value decomposition (SVD) (see examples in [Golub and Reinsch, 1971]) and the non-negative matrix factorization (NMF) [Lee and Seung, 1999], where the factorization is restricted to matrices with non-negative entries. This non-negativity in resulting factor matrices U and V allows interpretation of the results. Binary matrix factorization (BMF) [Zhang et al., 2007, Zhang et al., 2010] is a variant rooted from NMF where factor matrices are binary, while probabilistic nonnegative matrix factorization (PMF) [Laurberg et al., 2008, Gaussier and Goutte, 2005] approximates data as samples from a multinomial distribution.

The Cancer algorithm [Karaev and Miettinen, 2016a] works with continuous data, performing subtropical/tropical matrix factorization (SMF) on the input matrix, and returning two factorized matrices. The algorithm's two key ideas are: iteratively updating the rank-1 factors one-by-one and approximating the reconstruction error with a polynomial of low-degree. Latitude algorithm [Karaev et al., 2018] combines NMF and SMF, where factors are interpreted as NMF or SMF features or as mixtures of both. Unfortunately, neither Cancer not Latitude do not guarantee the convergence of the algorithms. Also, the authors [Weston et al., 2013] used subtropical semiring as part of a recommender system, which can be considered as a special kind of neural network.

De Schutter & De Moor introduced in 1997 a heuristic algorithm [De Schutter and De Moor, 1997] to compute factorization of a matrix in the tropical semiring, which we denote as Tropical Matrix Factorization (TMF). They use it to determine the minimal system order of a discrete event system (DES). In the last decades, there has been an increase in this research area, and DES is modeled as a max-plus-linear (MPL) system.

To implement TMF we need to know how to solve tropical linear systems. A tropical linear system is not solvable in general. For $A = [a_{ij}] \in \mathbb{R}_{\max}^{m \times n}$ and $c = [c_k] \in \mathbb{R}_{\max}^m$ we call the solutions $x \in \mathbb{R}_{\max}^n$ of the inequality

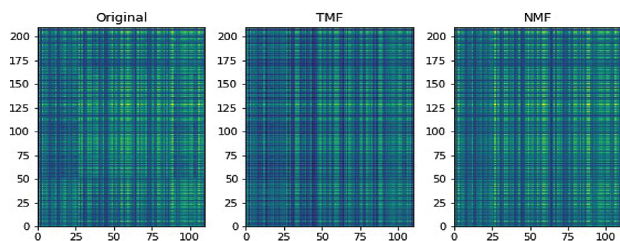
$A \otimes x \leq c$ the subsolutions of the linear system $A \otimes x = c$. The greatest subsolution $x = [x_1 \ x_2 \ \dots \ x_n]^T$ of $A \otimes x \leq c$ can be computed by

$$x_i = \max_{1 \leq j \leq m} (c_j - a_{ji}),$$

for $i = 1, 2, \dots, n$, where symbol \max denotes the standard subtraction in \mathbb{R} [Gaubert and Plus, 1997].

TMF starts with an initial guess for the matrix U in (1), denoted by U_0 , and then computes V as the greatest subsolution X of the equation $U_0 \otimes X = A$. Then authors use the iterative procedure by selecting and adapting an entry of U or V and recomputing it as the greatest subsolutions of $Y \otimes V = A$ and $U \otimes X = A$, respectively. The b -norm defined as $\|A_b\| = \sum_{i,j} |a_{ij}|$ is used as the objective function to get a good approximation of the input data.

In contrast to Cancer and Latitude, TMF update rules gradually reduce the approximation error and thus TMF algorithm is convergent. However, none of the existing tropical and subtropical algorithms

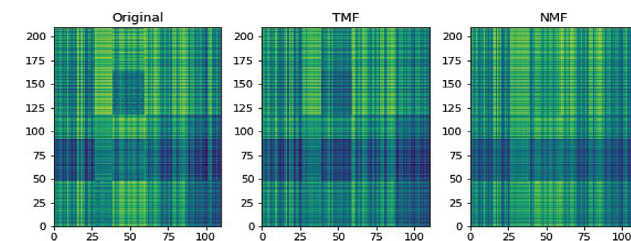


(a) Original synthetic matrix D_s and its two approximations. Correlation between TMF and NMF approximated matrices and matrix D_s is equal to 0.983 and 0.990, respectively.

Cancer and TMF, as defined, cannot be used for prediction tasks in data mining problems. Note that in TMF method there is no non-negativity constraint compared to the NMF and its variants. However, a weakness of TMF compared to NMF is in its computational efficiency.

4 RESULTS

We compare TMF and NMF on synthetic data created as a product of two non-negative random matrices. The objective of synthetic experiments is to show that the TMF can identify the $(\max, +)$ structure when it exists. Therefore, we construct two synthetic matrices: $D_s \in \mathbb{R}^{210 \times 110}$ as the standard product $(+, \times)$ of two random matrices of sizes 210×2 and 2×110 ; and $D_t \in \mathbb{R}^{210 \times 110}$ as the tropical product $(\max, +)$ of the same two matrices.



(b) Original synthetic matrix D_t and its two approximations. Correlation between TMF and NMF approximated matrices and matrix D_t is equal to 0.974 and 0.966, respectively.

Figure 1: Original and approximated matrices of rank 3, obtained by TMF and NMF.

As expected, NMF reconstructs the matrix D_s better as TMF, see Figure 1a. Results on Figure 1b show that NMF cannot successfully recover the patterns when dealing with specific synthetic data. Moreover, for matrix D_t TMF returns a better approximation as NMF.

5 CONCLUSION

Standard matrix factorization methods perform learning tasks over matrices equipped with addition and multiplication. The constructed models are linear and thus unable to model complex, non-linear relations. This can be addressed by introducing the tropical semiring with $(\max, +)$ operations.

To the best of our knowledge, we are the first to implement and apply TMF in data analysis. We showed that TMF gives better results than NMF, when the data is created by an underlying process of $(\max, +)$ semiring.

In our future work, we plan to adapt TMF to be able to predict missing values and test methods on real data. Because the resulting structure can be simpler to interpret than with standard linear algebra, we believe that future research will show that semirings are useful in many scenarios.

REFERENCES

- [1] De Schutter, B. and De Moor, B. (1997). Matrix factorization and minimal state space realization in the max-plus algebra. In *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, volume 5, pages 3136–3140. IEEE.
- [2] Gaubert, S. and Plus, M. (1997). Methods and applications of $(\max, +)$ linear algebra. In *Annual symposium on theoretical aspects of computer science*, pages 261–282. Springer.
- [3] Gaussier, E. and Goutte, C. (2005). Relation between pls and nmf and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602. ACM.

- [4] Golub, G. H. and Reinsch, C. (1971). Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer.
- [5] Karaev, S., Hook, J., and Miettinen, P. (2018). Latitude: A model for mixed linear-tropical matrix factorization. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 360–368. SIAM.
- [6] Karaev, S. and Miettinen, P. (2016a). Cancer: Another algorithm for subtropical matrix factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 576–592. Springer.
- [7] Karaev, S. and Miettinen, P. (2016b). Capricorn: An algorithm for subtropical matrix factorization. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 702–710. SIAM.
- [8] Karaev, S. and Miettinen, P. (2019). Algorithms for approximate subtropical matrix factorization. *Data Mining and Knowledge Discovery*, 33(2):526–576.
- [9] Laurberg, H., Christensen, M. G., Plumbley, M. D., Hansen, L. K., and Jensen, S. H. (2008). Theorems on positive data: On the uniqueness of nmf. *Computational intelligence and neuroscience*, 2008.
- [10] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788.
- [11] Weston, J., Weiss, R. J., and Yee, H. (2013). Nonlinear latent factorization by embedding multiple user interests. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 65–68.
- [12] Zhang, L., Naitzat, G., and Lim, L.-H. (2018). Tropical geometry of deep neural networks. *arXiv preprint arXiv:1805.07091*.
- [13] Zhang, Z., Li, T., Ding, C., and Zhang, X. (2007). Binary matrix factorization with applications. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 391–400. IEEE.
- [14] Zhang, Z.-Y., Li, T., Ding, C., Ren, X.-W., and Zhang, X.-S. (2010). Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, 20(1):28.
- [15] Žitnik, M. and Zupan, B. (2015). Data fusion by matrix factorization. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):41–53.

■

Amra Omanović received a master's degree from the Faculty of Computer and Information Science, University of Ljubljana, in 2018. She is a junior researcher at the Faculty of Computer and Information Science, University of Ljubljana. Her research is focused on the application of linear algebra over semirings in data embedding and fusion methods.

■

Polona Oblak received the doctoral degree from the Faculty of Mathematics and Physics, University of Ljubljana, in 2008. She is an associate professor at the Faculty of Computer and Information Science, University of Ljubljana. Her research interests include inverse eigenvalue problems for graphs, linear algebra over semirings and combinatorial matrix theory

■

Tomaž Curk received a doctoral degree from the Faculty of Computer and Information Science, University of Ljubljana, in 2007. He is an assistant professor and serves as a vice-dean for research at the Faculty of Computer and Information Science, University of Ljubljana. His research is focused on the application of machine learning and data integration methods in bioinformatics.

▣ Poglobljen pogled v beločnično biometrijo: nova podatkovna množica in študija delovanja

Matej Vitek¹, Peter Rot^{1,2}, Vitomir Štruc², Peter Peer¹

¹Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, Ljubljana, Slovenija

²Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, Ljubljana, Slovenija

{matej.vitek,peter.rot,peter.peer}@fri.uni-lj.si, vitomir.struc@fe.uni-lj.si

Izveček

Prepoznavanje ljudi je temeljni problem, s katerim se ukvarja področje biometrije. V našem delu se ukvarjamo s prepoznavo beločničnih žilnih struktur, ki imajo številne prednosti pred ostalimi značilnostmi: beločnične žile so edinstvene, tudi med identičnimi dvojčki – celo bolj kot prstni odtisi; za zajem ne potrebujemo posebnih naprav, le običajen fotoaparati ali mobilno kamero; zajem je neinvaziven in omogoča zajem na daljavo; žilne strukture se bistveno ne spreminjajo med življenjem; žilne strukture je težko ponarediti. Ker se biometrija, podobno kot mnoga ostala področja računalniškega vida, vedno bolj poslužuje globokega učenja, so izredno pomembne kvalitetne dobro označene podatkovne množice. Naša podatkovna množica SBVPI je javno dostopna in vsebuje očesne slike visoke kvalitete skupaj z ročnimi oznakami različnih regij v očesu (šarenica, zenica, ...). Poleg predstavitve SBVPI pa se naše delo tudi poglobi v bolj splošna vprašanja beločnične biometrije. V eksperimentalnem delu primerjamo 5 različnih pristopov za prepoznavo, ki jih evalviramo na SBVPI. Poleg tega analiziramo vpliv ločljivosti slik in smeri pogleda na uspešnost teh pristopov. Globoki model se je izkazal za najboljšega v splošnih poskusih, obenem pa tudi za najbolj robustnega, saj se je najbolje odrezal pri manjših ločljivostih in manj smereh pogleda. S tako poglobljeno analizo smo naslovili številna odprta vprašanja v beločnični biometriji, predvsem ker prikazujejo smiselnost uporabe globokega učenja (ki jo omogoča naša podatkovna množica SBVPI) in pa pomembnost uporabe različnih smeri pogleda v sami prepoznavi.

Ključne besede: Podatkovna množica, Prepoznavanje oseb, Očesna biometrija, Beločnica, Žilna biometrija.

Abstract

Ocular biometrics is the study of the applicability of various ocular modalities in different tasks, most prominently identity recognition. It can be useful in various applications such as surveillance systems, forensics or authentication systems. However, the existing datasets for ocular research are often inappropriate for the study of all three of the visible modalities – the sclera, the iris and the periocular region. In this work, we present a novel dataset of high-quality eye images captured in the visible spectrum appropriate for the study of all three modalities. We have performed an analysis of the covariance with several state-of-the-art recognition methods, studying the performance of the approaches not only on the dataset itself but also across different image resolutions and gaze directions. The results of this comprehensive study give insight not only into the general usability of our dataset, but also into the effects of different image resolutions and gaze directions on the accuracy of sclera-based recognition methods. Our experiments show that deep networks outperform handcrafted approaches in sclera recognition both in terms of overall performance as well as the robustness to lower resolutions and missing gaze directions.

Keywords: Dataset, identity recognition, ocular biometrics, sclera, vein-pattern recognition.

1 UVOD

Očesna biometrija je že od Daugmanovega dela na temo razpoznave šarenice [1] aktivno področje biometričnih raziskav in velja za eno najbolj natančnih

nih vej biometrije s premnogimi primeri uporabe v različnih domenah [2]. Večina rabljenih podatkovnih množic v literaturi pa ni namenjena beločničnim raziskavam in so



Slika 1: Primer slike iz SBVPI in pripadajočih oznak: beločnice, periokularne regije, šarenice, beločničnih žil, zenice, trepalnic, kantusa.

zato zajete v bližnjem infrardečem (NIR) področju ali pa nimajo potrebnih oznak beločnice in žil, ki so potrebne za raziskovanje (predvsem globokih) pristopov za segmentacijo in prepoznavo beločnice. Da bi naslovili te pomanjkljivosti, v našem delu predstavimo naslednje doprinose:

Predstavimo novo javno dostopno očesno podatkovno množico, ki je namenjena razpoznavi beločnice (vseeno pa je primerna tudi za ostale očesne modalnosti), z imenom SBVPI. Množica vsebuje kvalitetne slike in ročne označbe regij očesa, vključno z ročno označenimi žilami pri podmnožici slik (česar nima nobena druga javno dostopna množica).

Evalviramo in primerjamo pet sodobnih pristopov za razpoznavo na naši množici. S tem pokaže-

mo, da je množica dovolj kvalitetna za pomenljive rezultate, obenem pa dovolj težka, da je zanimiva za nadaljnje raziskave.

S kovariančno analizo primerjamo delovanje petih pristopov iz prejšnje točke z manjšimi slikami in različnimi smerni pogleda.

2 PODATKOVNE MNOŽICE

V tabeli 1 pregledamo obstoječe podatkovne množice za očesno biometrijo in prikažemo razlike z našo SBVPI. Večina beločničnih raziskav uporablja UBIRIS v1 [3, 4, 5, 6, 7, 8, 9], MASD [10, 11, 12], UBIRIS v2 [13, 14], MICHE [13, 14] in UTIRIS [3]. Prav tako so pogosto uporabljene zasebne podatkovne množice, ki pa jih v tabeli ne naštevamo. SBVPI ima številne

Tabela 1: Obstoječe podatkovne množice na temo očesne biometrije.

Podatkovna množica	Modalnost†	Javna	Orodje za zajem	Spekter	Velikost slik	Št. oseb	Št. slik	Oznake beločnice	Oznake žil	Smer(i) pogleda
CASIA Iris v1 [15]	Š	Da	Lasten skener šarenice	NIR	320 × 280	54	756	Ne	Ne	Statična
CASIA Iris v2 [15]	Š	Da	Lasten skener šarenice	NIR	640 × 480	60	2400	Ne	Ne	Statična
CASIA Iris v3 [15]	Š	Da	Lasten & OKI skener šarenice	NIR	640 × 480	> 700	22 034	Ne	Ne	Statična
CASIA Iris v4 [15]	Š	Da	Lasten & OKI skener šarenice	NIR	640 × 480	> 2800	54 601	Ne	Ne	Statična
ND-IRIS-0405 [16]	Š	Da	LG 2200 IIS	NIR	640 × 480	356	64 980	Ne	Ne	Statična
UPOL [17]	Š	Da	TOPCON TRC50IA	VIS	768 × 576	64	384	Ne	Ne	Statična
UTIRIS [18]	BŠ	Da	Canon EOS 10D	Oba	2048 × 1360	79	1540	Ne	Ne	Statična
UBIRIS v1 [19]	BŠ	Da	Nikon E5700	VIS	800 × 600	241	1877	Ne	Ne	Statična
UBIRIS v2 [20]	BPŠ	Da	Canon EOS 5D	VIS	400 × 300	261	11 102	Ne	Ne	Različne
ICE [21]	Š	Ne	N/A	N/A	N/A	120	3056	Ne	Ne	N/A
WVU [22]	Š	Ne	N/A	Oba	N/A	380	1852	Ne	Ne	N/A
IITD [23]	Š	Da	JIRIS JPC1000 CMOS Camera	NIR	320 × 240	224	1120	Ne	Ne	Statična
MICHE-I [24]	BPŠ	Da	Mobilna kamera	VIS	2048 × 1536	92	3732	Ne	Ne	Statična
UBIPr [25]	P	Da	Canon EOS 5D	VIS	500 × 400	261	10 950	Ne	Ne	Različne
UBIPosePr [26]	P	Da	N/A	VIS	N/A	100	2400	Ne	Ne	Različne
IMP [27]	P	Da	Cogent Iris Scanner, Nikon SLR	Oba	260 × 270	62	930	Ne	Ne	Statična
IUPUI [28]	SPI	Ne	N/A	Oba	N/A	44	352	Ne	Ne	Različne
MASD [29]	S	Ne	NIKON D 800	VIS	7500 × 5000	82	2624	Podmnožica	Ne	Različne
SBVPI (naša)	SPI	Da	Canon EOS 60D	VIS	3000 × 1700	55	1858	Celota	Podmnožica	Različne

† B – Beločnica, P – Periokularna regija, Š – šarenica

Oznaka N/A pomeni, da informacije nismo mogli pridobiti iz podatkovne množice in pripadajoče literature.

¹ SBVPI je javno dostopna za raziskovalne namene na: <http://sclera.fri.uni-lj.si/>

prednosti pred obstoječimi podatkovnimi množicami, saj je edina javno dostopna očesna podatkovna množica z ročno označenimi beločnicami in žilami.

V tabeli 2 predstavimo glavne lastnosti SBVPI1. V sliki 1 pokažemo primer slike SBVPI in pripadajočih oznak očesnih regij.

Zaradi oznak očesnih regij in metapodatkov, ki so vključeni v podatkovni množici, je SBVPI, v nasprotju z obstoječimi podatkovnimi množicami,

primerna za raziskavo pristopov za segmentacijo in prepoznavo beločnice. Na piksele natančne oznake žil na podmnožici slik so še posebej pomembne in so edinstvena lastnost podatkovne množice SBVPI. Ker SBVPI vsebuje tudi oznake preostalih regij, je primerna tudi za študijo pristopov razpoznave šarenice in periokularne regije v vidnem delu spektra, kar omogoča študijo pristopov združenja različnih očesnih modalnosti.

Tabela 2: Opis podatkovne množice SBVPI (Sclera Blood Vessels, Periocular and Iris).

Lastnost	Opis
Število slik	1858
Število oseb	55
Starost oseb	15–80, večina pod 35
Spoli oseb	29 žensk, 26 moških
Število slik na osebo	Najmanj 32, nekatere osebe več
Velikost slik	3000 × 1700 pikslov
Oznake (vse slike)	Beločnica, periokularna regija
Oznake (~ 130 slik)	Šarenica, zenica, kantus, trepalnice, ožilje
Metapodatki (vse slike)	Identiteta, levo/desno oko, smer pogleda, starost, spol, barva
Smeri pogleda	Levo, desno, navzgor, naravnost

3 PREPOZNAVA BELOČNICE

V našem delu primerjamo klasične pristope, ki so se do nedavnega uporabljali v biometriji in globoko učenje, ki predstavlja smer, v katero se v zadnjem času premika računalniški vid v splošnem. Uporabimo SegNet iz [30] (ki je bil naučen z žilnimi oznakami iz SBVPI in je trenutno edini model za segmentacijo žil beločnice v literaturi), da iz slik izločimo žilne maske. Nato implementiramo globoki model ScleraNet [30] in ga na problemu razpoznave identitete primerjamo z deskriptorskimi metodami, ki so se uporabljale pred globokim učenjem: tremi ki temeljijo na detekciji ključnih točk (SIFT [31], SURF [32], and ORB [33]) in gostem SIFTu [34], ki temelji na gosti mreži. Vse metode poganjamo na žilnih maskah prej omenjenega modela SegNet, da omogočimo pravično primerjavo razpoznavnih metod.

V verifikacijskih eksperimentih uporabimo galerijo (ki vsebuje 4 naključne slike za vsako osebo – 1 za vsako smer pogleda) in poskusno množico (ki vsebuje vse ostale slike). V enem poskusu avtentikacije

primerek iz poskusne množice primerjamo z vsemi 4 iz galerije s pristopi za prepoznavo, ki smo jih opisali zgoraj. Najmanjšo od 4 razdalj vzamemo kot ujemanje, na podlagi katerega potem avtentikacijo sprejmemo ali zavrnemo. Ta odločitev je lahko pravilna ali napačna, glede na to, ali je oseba iz poskusne množice enaka galerijski ali ne. S spreminjanjem praga sprejetja lahko nadziramo ravnovesje med deležem napačnih sprejetij (angl. false acceptance rate – FAR) in deležem napačnih zavrnitev (angl. false rejection rate – FRR). Rezultat lahko poročamo kot površino pod krivuljo ROC, ki slika to ravnovesje pri različnih pragovih (angl. area under curve – AUC). V literaturi se uporablja tudi metrika točke enake napačnosti (angl. equal error rate – EER), ki predstavlja točko, kjer sta FAR in FRR enaka. Ker so napačna sprejetja bolj kritična kot napačne zavrnitve, poročamo tudi delež verifikacije ($VER = 1 - FRR$) pri nizkih vrednostih FAR (v našem delu 1 % in 0.1 %). V tabeli 3 prikazemo rezultate splošnih poskusov.

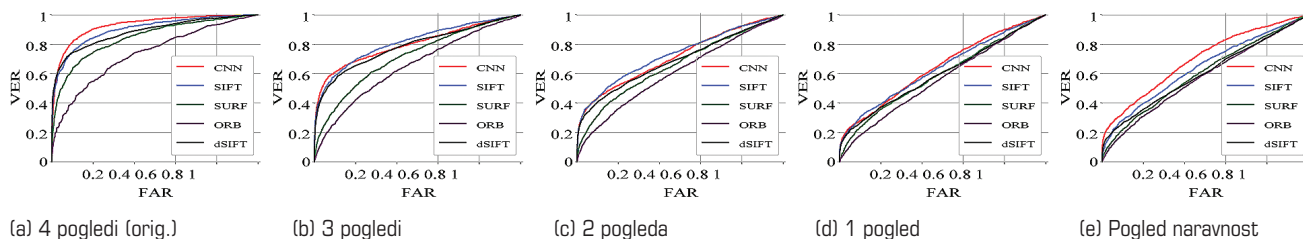
Tabela 3: Rezultati splošne prepoznave, v katerih vidimo, da se je ScleraNet (označen s CNN) odrezal najbolje v celovitem delovanju, medtem ko je bil gosti SIFT (označen z dSIFT) najboljši pri nizkih vrednostih FAR. Rezultati so podani kot $\mu \pm \sigma$ (z 10-kratno evalvacijo na SBVPI).

Model	VER@0.1FAR	VER@1FAR	EER	AUC
CNN	0.178 ± 0.008	0.451 ± 0.008	0.144 ± 0.003	0.933 ± 0.003
SIFT	0.184 ± 0.076	0.452 ± 0.040	0.176 ± 0.005	0.903 ± 0.005
SURF	0.100 ± 0.008	0.285 ± 0.014	0.233 ± 0.006	0.844 ± 0.005
ORB	0.052 ± 0.011	0.148 ± 0.013	0.330 ± 0.007	0.736 ± 0.007
dSIFT	0.256 ± 0.012	0.496 ± 0.012	0.193 ± 0.003	0.889 ± 0.004

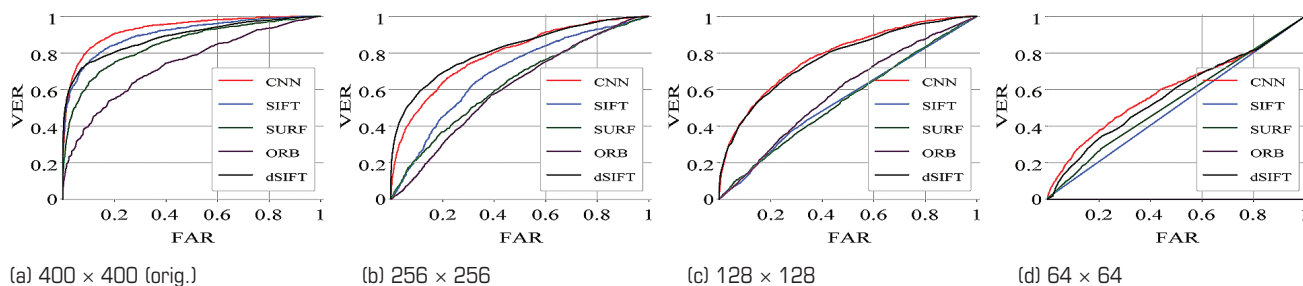
Ker se v različnih smereh pogleda vidijo različne žilne strukture, smo v galerijo vedno vključili vse 4 smeri pogleda. Zanima pa nas, kako bi se rezultati spremenili, če bi uporabili manj smeri pogleda. V drugi seriji poskusov zato galerijo s 4 pogledi nadomestimo z galerijo, kjer ima vsaka oseba le 3/2/1 slike, še vedno z različnimi (a naključno izbranimi) smermi pogleda. Ker pogled naravnost da najbolj ce-

lovito sliko žilnih struktur v očesu, poskus izvedemo še z galerijo s pogledom naravnost. Pri teh eksperimentih modela ScleraNet ne učimo ponovno, pač pa uporabimo enakega, kot smo ga uporabili za splošne poskuse.

Rezultate eksperimentov prikažemo v sliki 2. Več eksperimentov z bolj poglobljeno analizo je bilo opravljeno v razširjenem delu[35].



Slika 2: Rezultati poskusov z manjšimi galerijami. Prvi trije eksperimenti (b–d) v galeriji vsebujejo 3/2/1 smeri pogleda za vsako osebo, zadnji pa ima v galeriji samo pogled naravnost. ScleraNet (označen s CNN) in SIFT se obneseta najbolje pri naključnih smerih, pri pogledu naravnost pa ScleraNet opazno premaga vse ostale pristope. ScleraNet tudi edini opazno izboljša delovanje pri prehodu iz ene naključne smeri na pogled naravnost, kar pomeni, da se najbolje nauči pomena celotnega očesnega ožilja.



Slika 3: Rezultati poskusov s pomanjšanimi slikami. ScleraNet (označen s CNN) in gosti SIFT (označen z dSIFT) se izkažeta kot precej robustna, saj oba delujeta dobro na velikostih do 128 × 128. ScleraNet pa opazno premaga ostale pristope tudi na manjših slikah, saj očitno najde pomenljive značilke celo na slikah velikosti 64 × 64.

Zanima nas tudi potrebna kvaliteta slik in robustnost pristopov na manj kvalitetne slike, zato slike velikosti 400 × 400, ki smo jo uporabili v prejšnjih eksperimentih, pomanjšamo na različne manjše ločljivi-

vosti. Pri tem spet uporabimo enak model ScleraNet in ga ne učimo še enkrat na manjših slikah. Rezultate eksperimentov poročamo v sliki 3.

4 SKLEP

Predstavili smo novo podatkovno množico namenjeno raziskovanju beločnične biometrije in pokazali njene prednosti pred ostalimi obstoječimi javnimi podatkovnimi množicami, ki se uporabljajo za očesno biometrijo. Med pomembnejšimi prednostmi so: javna dostopnost, kvalitetne slike visoke ločljivosti, zajetje v vidnem delu spektra in (najpomembnejše) ročne označbe očesnih regij, vključno s podrobnimi označbami očesnih žil.

Evalvirali in primerjali smo pet sodobnih pristopov za prepoznavo – en globok model, tri deskriptorske metode z detektorjem ključnih točk in eno deskriptorsko metodo z gosto mrežo. Naši rezultati prikazujejo, da globoki model deluje bolje kot deskriptorske metode, kar je v skladu z rezultati zadnjih let v biometriji in ostalih področjih računalniškega vida. Gosti SIFT pa je deloval najbolje pri nizkih vrednostih FAR.

Raziskali smo tudi vpliv različnih smeri pogleda in velikosti slik na delovanje razpoznavne beločnice. Pomankljive galerije z manj smermi pogleda povzročijo precej slabše delovanje, saj se pri različnih smereh pogleda vidijo različne žilne strukture. Globoki model se je izkazal za najbolj robustnega pri manjših galerijah, obenem pa je bil edini, ki je pri pogledu naravnost deloval bolje kot pri 1 naključno izbrani smeri pogleda, kar nakazuje, da se je najbolje naučil celovite predstavitve žilnih struktur (saj se le-te pri pogledu naravnost vidijo najbolj celovito). Gosti SIFT in globoki model sta se izkazala za najbolj odporna na slabšo kvaliteto slik, saj sta dobro delovala na slikah vse do velikosti 128×128 , globoki model pa je pomenljive predstavitve žil našel celo na slikah do velikosti 64×64 in se tako ponovno izkazal za najbolj robustnega.

Z najboljšim rezultatom točke enake napake 14,4 % in precej nizkih verifikacijskih uspešnostih pri nizkih vrednostih FAR ostaja precej prostora za izboljšave tudi na samem pristopu za prepoznavo, saj bi za praktično uporabo želeli boljše rezultate z majhnim številom napačnih sprejetij.

LITERATURA

- [1] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.
- [2] I. Nigam, M. Vatsa, and R. Singh, "Ocular biometrics: A survey of modalities and fusion approaches," *Information Fusion*, vol. 26, pp. 1–35, 2015.
- [3] S. Alkassar, W. L. Woo, S. S. Dlay, and J. A. Chambers, "Robust sclera recognition system with novel sclera segmentation and validation techniques," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 474–486, 2017.
- [4] A. Das, U. Pal, M. A. F. Ballester, and M. Blumenstein, "A new efficient and adaptive sclera recognition system," in *2014 IEEE Symposium on Computational Intelligence in Biometrics and Identity Management (CIBIM)*, pp. 1–8, IEEE, 2014.
- [5] V. Gottemukkula, S. Saripalle, S. P. Tankasala, and R. Derakhshani, "Method for using visible ocular vasculature for mobile biometrics," *IET Biometrics*, vol. 5, no. 1, pp. 3–12, 2016.
- [6] Y. Lin, E. Y. Du, Z. Zhou, and N. L. Thomas, "An efficient parallel approach for sclera vein recognition," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 147–157, 2014.
- [7] K. Oh and K. A. Toh, "Extracting sclera features for cancelable identity verification," in *2012 5th IAPR International Conference on Biometrics (ICB)*, pp. 245–250, IEEE, 2012.
- [8] N. L. Thomas, Y. Du, and Z. Zhou, "A new approach for sclera vein recognition," in *Mobile Multimedia/Image Processing, Security, and Applications 2010*, vol. 7708, p. 770805, International Society for Optics and Photonics, 2010.
- [9] Z. Zhou, E. Y. Du, N. L. Thomas, and E. J. Delp, "A new human identification method: Sclera recognition," *IEEE Transactions on Systems, Man, and Cybernetics (TSMC) -Part A: Systems and Humans*, vol. 42, no. 3, pp. 571–583, 2012.
- [10] P. Radu, J. Ferryman, and P. Wild, "A robust sclera segmentation algorithm," in *International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, pp. 1–6, 2015.
- [11] D. Riccio, N. Brancati, M. Frucci, and D. Gragnaniello, "An unsupervised approach for eye sclera segmentation," in *Iberoamerican Congress on Pattern Recognition (CIARP)*, pp. 550–557, Springer, 2017.
- [12] P. Rot, V. Štruc, and P. Peer, "Deep multi-class eye segmentation for ocular biometrics," in *IEEE International Work Conference on Bioinspired Intelligence*, 2018.
- [13] S. Alkassar, W. L. Woo, S. Dlay, and J. Chambers, "Sclera recognition: on the quality measure and segmentation of degraded images captured under relaxed imaging conditions," *IET Biometrics*, vol. 6, no. 4, pp. 266–275, 2016.
- [14] D. R. Lucio, R. Laroca, E. Severo, A. S. B. Jr, and D. Menotti, "Fully convolutional networks and generative adversarial networks applied to sclera segmentation," in *IEEE ICB: Theory, Applications, and Systems*, 2018.
- [15] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient iris recognition by characterizing key local variations," *IEEE Transactions on Image processing*, vol. 13, no. 6, pp. 739–750, 2004.
- [16] K. W. B. Bowyer and P. J. Flynn, "The ND-IRIS-0405 iris image dataset," tech. rep., Notre Dame University.
- [17] M. Dobeš, L. Machala, P. Tichavský, and J. Pospíšil, "Human eye iris recognition using the mutual information," *Optik*, vol. 115, no. 9, pp. 399–404, 2004.
- [18] M. S. Hosseini, B. N. Araabi, and H. Soltanian-Zadeh, "Pigment melanin: Pattern for iris recognition," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, pp. 792–804, april 2010.
- [19] H. Proença and L. Alexandre, "Ubiris: A noisy iris image database," in *International Conference on Image Analysis and Processing*, pp. 970–977, Springer, 2005.
- [20] H. Proença, S. Filipe, R. Santos, J. Oliveira, and L. A. Alexandre, "The ubiris. v2: A database of visible wavelength iris images captured on-the-move and at-a-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1529–1535, 2010.

- [21] P. J. Phillips, K. W. Boyer, P. J. Flynn, A. J. O'Toole, P. J. Phillips, C. L. Schott, W. T. Scruggs, and M. Sharpe, "Frvt 2006 and ice 2006 large-scale results," 2007.
- [22] C. Whitelam, Z. Jafri, and T. Bourlai, "Multispectral eye detection: A preliminary study," in *2010 20th International Conference on Pattern Recognition*, pp. 209–212, IEEE, 2010.
- [23] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern Recognition*, vol. 43, no. 3, pp. 1016 – 1026, 2010.
- [24] M. D. Marsico, M. Nappi, D. Riccio, and H. Wechsler, "Mobile iris challenge evaluation (miche)-i, biometric iris dataset and protocols," *Pattern Recognition Letters*, vol. 57, pp. 17 – 23, 2015. Mobile Iris CHallenge Evaluation part I (MICHE I).
- [25] C. N. Padole and H. Proença, "Periocular recognition: Analysis of performance degradation factors," in *Biometrics (ICB), 2012 5th IAPR International Conference on*, pp. 439–445, IEEE, 2012.
- [26] C. N. Padole and H. Proença, "Compensating for pose and illumination in unconstrained periocular biometrics," *International Journal of Biometrics*, vol. 5, no. 3-4, pp. 336–359, 2013.
- [27] A. Sharma, S. Verma, M. Vatsa, and R. Singh, "On cross spectral periocular recognition," in *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 5007–5011, IEEE, 2014.
- [28] Z. Zhou, E. Y. Du, N. L. Thomas, and E. J. Delp, "Multi-angle sclera recognition system," in *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2011 IEEE Workshop on*, pp. 103–108, IEEE, 2011.
- [29] A. Das, U. Pal, M. A. Ferrer, M. Blumenstein, D. Štepec, P. Rot, Ž. Emeršič, P. Peer, V. Štruc, S. Kumar, et al., "SSERBC 2017: Sclera segmentation and eye recognition benchmarking competition," in *International Joint Conference on Biometrics (IJCB)*, pp. 742–747, 2017.
- [30] P. Rot, M. Vitek, K. Grm, Ž. Emeršič, P. Peer, and V. Štruc, "Deep sclera segmentation and recognition," in *Handbook of Vascular Biometrics* (A. Uhl, C. Busch, S. Marcel, and R. Velthuis, eds.), pp. 395–432, Springer, 2020.
- [31] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Journal of Computer Vision (ICCV)*, vol. 2, pp. 1150–1157, IEEE, 1999.
- [32] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, pp. 404–417, Springer, 2006.
- [33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *International conference on computer vision*, pp. 2564–2571, 2011.
- [34] A. Das, U. Pal, M. A. F. Ballester, and M. Blumenstein, "Sclera recognition using dense-sift," in *2013 13th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 74–79, IEEE, 2013.
- [35] M. Vitek, P. Rot, V. Štruc, and P. Peer, "A comprehensive investigation into sclera biometrics: a novel dataset and performance study," *Neural Computing and Applications*, pp. 1–15, 2020.

■

Matej Vitek je mladi raziskovalec in doktorski študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Leta 2015 je prejel diplomu s strani Fakultete za računalništvo in informatiko ter Fakultete za matematiko in fiziko, leta 2018 pa je od istih dveh ustanov prejel še magisterij. Trenutno se ukvarja z (predvsem lahкими modeli za) biometrijo beločnice, deloval pa je tudi na področju umetne inteligence in razvoja iger.

■

Peter Rot je doktorski študent na Univerzi v Ljubljani na Fakulteti za računalništvo in informatiko. Leta 2015 je diplomiral iz računalništva in informatike, leta 2018 pa magistriral, prav tako iz računalništva in informatike. Trenutno se ukvarja z ohranjanjem zasebnosti in generiranjem obrazov, v preteklosti pa je deloval tudi na področju očesne biometrije.

■

Vitomir Štruc je izredni profesor na Fakulteti za elektrotehniko na Univerzi v Ljubljani. Od te fakultete je prejel tudi doktorat leta 2010. Vitomir je mednarodno priznan strokovnjak na področjih računalniškega vida, obdelave slik, prepoznavanja vzorcev in strojnega učenja. Je avtor (ali soavtor) čez 100 raziskovalnih člankov za priznane mednarodne znanstvene revije in konference na teh področjih. Pomagal je tudi pri organizaciji številnih svetovnih konferenc na temo računalniškega vida, med njimi IEEE Face and Gesture, ICB, WACV. Trenutno je predsednik programskega odbora IJCB 2020. Vitomir je pridružen urednik IEEE Transactions on Information Forensics and Security, Pattern Recognition, Signal Processing in IET Biometrics. Bil je meta recenzent za WACV 2018, 2019, 2020, ICPR 2018, Eusipco 2019 in FG 2020. Je član IEEE, IAPR, EURASIP, trenutni predsednik SDRV in slovenski kontakt za EAB.

■

Peter Peer je redni profesor na Fakulteti za računalništvo in informatiko na Univerzi v Ljubljani, kjer je tudi vodja Laboratorija za računalniški vid, koordinator dvojnega študijskega programa z južnokorejsko univerzo Kyungpook National University in prodekan za gospodarske zadeve. Na Fakulteti za računalništvo in informatiko je doktoriral leta 2003. Po doktoratu je bil vabljen podoktorski raziskovalec na CEIT v San Sebastianu v Španiji. Raziskovalno se ukvarja z računalniškim vidom s poudarkom na biometriji. Sodeloval je v številnih državnih in EU projektih in je avtor več kot 100 raziskovalnih člankov v priznanih revijah in na konferencah. Nedavno je bil soorganizator tekmovanja Unconstrained Ear Recognition Challenge na IEEE/IAPR International Joint Conference on Biometrics 2017 in IAPR International Conference on Biometrics 2019. Leta 2018 je bil na to temo tudi sourednik posebne izdaje v reviji IET Biometrics. Trenutno je področni urednik v revijah IEEE Access in IET Biometrics. Je član EAB, IAPR in IEEE, v slovenski sekciji IEEE pa je 4 leta bil tudi predsednik računalniškega poglavja.

■ Digitalizacija in prilagoditev psihološkega testa za uporabo s sistemom za spremljanje očesnih gibov

Vida Groznik^{1,2,3}

¹Univerza na Primorskem, Fakulteta za matematiko, naravoslovje in informacijske tehnologije, Glagoljaška 8, 6000 Koper

²Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana

³NEUS Diagnostics, d.o.o., Zihlerova ulica 40E, 1000 Ljubljana
vida.groznik@famnit.upr.si

Izvleček

Demenca je najpogostejša nevrološka motnja, za katero po svetu boleha okoli 50 milijonov ljudi, ta številka pa se zaradi staranja prebivalstva vztrajno veča. Raziskave kažejo, da se znaki demence v obliki kognitivnega upada nakazujejo že nekaj let pred jasno izraženo klinično sliko. V primeru, da te znake odkrijemo v fazi blagega kognitivnega upada (MCI), lahko s različnimi pristopi upočasnimo ali celo zaustavimo napredovanje bolezni v demenco. Za postavitve diagnoze je trenutno potreben pregled pri nevrologu in psihologu, ki postavi diagnozo. Žal pa le redki bolniki pridejo do specialistov, ko so še v fazi MCI. Eden izmed testov, ki se ga psihologi pogosto poslužujejo, je test s Corsijevimi kockami. S pomočjo le-tega psiholog lahko oceni stopnjo okvare vidno-prostorskega delovnega spomina, ki med drugim lahko nakazuje na MCI. Test se ponavadi izvaja s fizičnimi kockami, razporejenimi na podlago. Psiholog v naključnem zaporedju tapka po kockah, naloga preizkušanca pa je, da za psihologom ponovi tapkanje v istem vrstnem redu. Eden izmed načinov za pravočasno detekcijo MCI je digitalizacija in avtomatizacija nevropsihološkega testiranja z uporabo sodobnih tehnologij. V zadnjih letih so se močno izboljšali sistemi za sledenje očesnim gibom, zaradi njihove cenovne dostopnosti pa so postali zanimivo orodje za potrebe raziskovanja nevroloških bolezni, pogosto tudi za raziskovanje različnih stopenj kognitivnega upada. V članku predlagamo prilagoditev in digitalizacijo testa Corsi za uporabo skupaj s sistemom za sledenje očesnim gibom. Skupaj s še nekaterimi drugimi testi (npr. gladko sledenje točki) bi tako predstavljal baterijo nalog za avtomatsko detekcijo znakov MCI z uporabo sistema za sledenje očesnim gibom. Tovrsten sistem bi lahko kot neke vrste presejalni test uporabljalo zdravstveno varstvo na primarni ravni, ki bi ob odkritju prisotnosti MCI bolnika pravočasno napotil na pregled pri specialistu.

Ključne besede: blag kognitivni upad, digitalizacija psiholoških testov, sistem za sledenje očesnih gibov, test Corsi

Abstract

Dementia is the most common neurological disorder as around 50 million people worldwide suffer from it. Due to the aging of the population, this number is steadily increasing. Research shows that signs of dementia in the form of cognitive decline are present already several years before having a clearly expressed clinical picture. If these signs could be detected in the phase of mild cognitive impairment (MCI), it might be possible to slow down or even stop the progression of the disease to dementia via various approaches. Currently, neurological and psychological examinations are required to make the diagnosis of MCI. Unfortunately, only a small number of patients visit a specialist while still in the MCI phase. The Corsi block-tapping test is one of the tests that psychologists use to assess the degree of visuospatial working memory impairment that may, among other, indicate MCI. The test is usually performed with physical blocks arranged on the board in an irregular pattern. The psychologist taps the blocks in random order and the patient repeats the tapping in the same order. One way to detect MCI in a timely manner would be to digitize and automate neuropsychological testing using available modern technologies. In recent years, eye-tracking systems have been improved substantially and their affordability has made them interesting tools for the study of neurological diseases, including the study of varying degrees of cognitive decline. In this paper, we propose an adaptation and the digitization of the Corsi test for use with the eye-tracking technology. Along with certain other tests (e.g. smooth pursuit), it would be featured in a battery of neuropsychological tasks for the automatic detection of MCI using an eye-tracking system. Such a system could be used as a screening tool at the primary health care level. It would enable physicians to detect signs of MCI and refer such patients for further examination by a specialist.

Keywords: Mild cognitive impairment, digitization of psychological tests, eye-tracking system, Corsi test.

1 UVOD IN MOTIVACIJA

Zdravniki se pogosto srečujejo z osebami, ki izražajo zaskrbljenost nad dejstvom, da se njim ali njihovim bližnjim dogaja, da pogosteje pozabljajo. Ponavadi se ti pomisleki nanašajo na celo vrsto kognitivnih sposobnosti oz. upadom kognitivnih sposobnosti in ne nujno zgolj na spomin. V takšnih primerih morajo zdravniki naprej ugotoviti, ali gre za kognitivne spremembe, ki so značilne za normalno staranje ali gre pri tem za klinično pomembne spremembe. V primeru, da je kognitivna okvara tako huda, da ogroža bolnikovo normalno delovanje (npr. ni zmožen samostojno opravljati vsakodnevnih aktivnosti), govorimo o demenci. Stanju med demenco in normalnim kognitivnim delovanjem, pri čemer ima oseba ohranjene funkcionalne sposobnosti, pravimo blag kognitivni upad (*angl. Mild Cognitive Impairment (MCI)*). [Hugo and Ganguli, 2014]

Demenca je najbolj razširjeno nevrološko obolenje, za katero boleha okoli 50 milijonov ljudi po celem svetu. S staranjem prebivalstva se ta številka vztrajno povečuje. Vsako leto je potrjenih skoraj 10 milijonov novih primerov in pričakuje se, da bo v roku desetih let (torej do leta 2030) na svetu že 82 milijonov ljudi z demenco in kar 152 milijonov do leta 2050 [Alzheimer's Disease International, 2019]. Glede na to, da je MCI vmesna stopnja med normalnim kognitivnim delovanjem in demenco, ni presenetljivo, da približno 15 odstotkov bolnikov diagnosticiranih z MCI razvije Alzheimerjevo demenco ali parkinsonovo bolezen z demenco v roku dveh let.

Raziskave kažejo, da je moč z uporabo različnih pristopov zavreti ali celo preprečiti nastanek MCI oz. demence [Ayati et al., 2020]. Med te pristope sodijo različne terapije za nekatere druge bolezni, kot so različna nesteroidna protivnetna zdravila [Yip et al., 2005, Szekely et al., 2008] in statini [Swiger et al., 2013, Wong et al., 2013]. Za učinkovite so se izkazale različne nefarmaceutvske metode, kot na primer spremembe življenjskega sloga [Lourida et al., 2019], uživanje mediteranske prehrane [Grant, 1999, Féart et al., 2009, Solfrizzi et al., 2011, Lourida et al., 2013, Grant, 2016], zmanjšano uživanje alkohola [Letenneur, 2004], fizična telesna aktivnost [Hamer and Chida, 2009, Cass, 2017, Firth et al., 2018], pomembnost kvalitetnega spanca [Chang et al., 2013, Lim et al., 2013, Benedict et al., 2015] in socialnih stikov [Wang et al., 2002, Fratiglioni et al., 2004, Crooks et al., 2008]. Glede na različne ugotovitve znanstvenikov, da ob-

stajajo načini, da se zavre nastanek demence, je razumljivo, da se čedalje več energije vlaga v iskanje biomarkerjev za zgodnje odkrivanje demence in blagega kognitivnega upada.

Ocena kognitivnega stanja pacienta je v klinični praksi podana na podlagi nevropsihološke ocene specifičnih kognitivnih domen (pozornost, izvršilne funkcije, učenje in spomin, jezik, senzomotorika, socialna kognicija), ki omogoča razpoznavo tako blagih kognitivnih okvar kot tudi postavitvi diferencialne diagnoze [Hugo and Ganguli, 2014]. Kot pomoč pri podaji objektivne ocene se nevrologi in psihologi poslužujejo različnih globalnih presejalnih lestvic, kot so na primer dobro poznani Kratek preizkus spoznavnih sposobnosti (KPSS) [Granda et al., 2003, Rakuša et al., 2006] (*angl. Mini-Mental State Examination (MMSE)*) [Folstein et al., 1975]), Montrealska lestvica ocenjevanja spoznavnih sposobnosti (*angl. Montreal Cognitive Assessment (MoCA)*) [Nasreddine et al., 2005], Mini-Cog [Borson et al., 2000], Kognitivni preizkus Addenbrooke [Mioshi et al., 2006, McColgan et al., 2012] in drugi. Presejalne lestvice pogosto dopolnjujejo z različnimi testi (npr. Tower of London [de Paula et al., 2012, Rainville et al., 2012], Corsi Block-Tapping Test [Guariglia, 2007], risanje ure [Parsey and Schmitter-Edgecombe, 2011], ipd.), ki še dodatno doprinejajo k točnosti podane ocene.

Za postavitev končne ocene kognitivnega upada je navadno potreben pregled pri specialistu nevrologije, ki po potrebi pacienta napoti na poglobljeno psihološko testiranje, ki lahko traja nekaj ur, ter slikanje glave z magnetno resonanco. Kljub pomembnosti zgodnjega odkrivanja kognitivne okvare pa je s trenutnim načinom postavljanja diagnoze kadrovske, časovne in finančne nemogoče izvesti sistematična presejalna testiranja na širši populaciji ljudi. Če si želimo zmanjšati vpliv staranja prebivalstva na našo družbo, ki za seboj potegne tudi visoke stroške za zdravstveno blagajno in psihično ter fizično obremenjenost svojcev bolnika z demenco, je ena izmed možnih rešitev digitalizacija in avtomatizacija psihološkega testiranja, ki v tem trenutku vzame največ časa za postavitev diagnoze. S tem bi omogočili širši populaciji boljšo dostopnost do hitre ocene njihovega kognitivnega stanja in posledično primernega zdravljenja oz. spremembe življenjskih navad.

Prednosti digitaliziranih testov so predvsem v možnostih objektivne ocene stanja in spremljanje razvoja bolezni skozi čas, kar pri trenutnem načinu

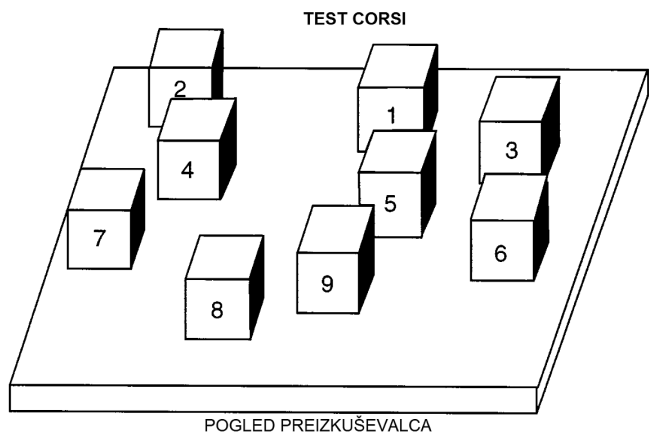
diagnosticiranja ni mogoče. Za razliko od klasičnih testiranj, preizkušanec rezultat testiranja lahko prejme takoj, ko s testiranjem zaključi in mu nanje ni potrebno čakati. Za izvedbo testiranja ni potrebna prisotnost specialista (nevrologije in/ali psihologije), temveč je dovolj prisotnost tehnika, ki zagotovi, da je test korektno izveden. S tem se tudi olajša dostopnost testa, saj ga lahko izvedejo v kateri koli zdravstveni ustanovi (s tem pa se preizkušancu prihrani morebitna dodatna pot v specializirano zdravstveno ustanovo). Ob dobri zasnovi digitaliziranega testa pa lahko olajšamo testiranje marsikateri osebi, ki testiranja zaradi fizičnega stanja trenutno ni zmožna opraviti. Sama digitalizacija nam poleg tega omogoča enostavno prilagajanje testa (z vidika nastavitve parametrov, diagnostičnega modela v ozadju, variacij testov ipd.), avtomatske analize in hitrejšo validacijo samega testa.

V nadaljevanju članka v drugem poglavju predstavimo enega izmed psiholoških testov - test Corsi in trenutni način administracije testa, v tretjem poglavju na kratko predstavimo sisteme za spremljanje očesnih gibov, ter nato v četrtem poglavju predstavimo način digitalizacije in prilagoditve testa Corsi, ki bi omogočal uporabo skupaj s sistemom za sledenje očesnim gibom. Članek zaključimo s sklepom in kratkim opisom nadaljnjega dela.

2 TEST S CORSIJEVIMI KOCKAMI

Test s Corsijevimi kockami (*angl. Corsi Block-Tapping Test*) (v nadaljevanju »test Corsi«) uporabljajo nevrologi in psihologi za oceno vidno-prostorskega kratkoročnega spomina. Leta 1972 je test razvil Philip Corsi [Corsi, 1972] in je postal eden najbolj pomembnih neverbalnih testov v nevropsihološkem raziskovanju [Berch et al., 1998], ki je med drugim vključen v baterijo testov WAIS-R [Kaplan, 1991]. Izvorni test Corsi je sestavljen iz devetih kock, ki so postavljene nepravilno na ploščo velikosti 23 x 28 cm, pri čemer so kocke oštevilčene s številkami od ena do devet. Shema postavitve kock je prikazana na sliki 1.

Testiranje izgleda tako, da preizkuševalec tapne kocke v naključnem zaporedju. Takoj za tem mora preizkušanec poskusiti ponoviti zaporedje tapkanja. Test se začne z zaporedjem dolžine dve, ki se nato tekom testiranja podaljšuje za eno dolžino do največ dolžine devet (kolikor je različnih kock). Vsaka dolžina zaporedja se ponovi dvakrat. Zaporedje se podaljšuje toliko časa, dokler je preizkušanec še



Slika 1: Postavitev kock na plošči pri testu Corsi z vidika preizkuševalca. (Slika je povzeta iz dela [Berch et al., 1998].)

zmožen pravilno ponoviti zaporedje, ki ga je tapnil preizkuševalec.

Izpeljanka osnovnega testa je takšna, da mora preizkušanec zaporedje ponoviti od kocke, ki jo je preizkuševalec tapnil zadnjo proti kocki, ki jo je preizkuševalec tapnil prvo. Osnovni test Corsi ima veliko izpeljank, ki se nanašajo na velikosti, zaporedja, postavitve in število kock (glej [Kessels et al., 2000, Berch et al., 1998]).

Poleg testa Corsi v fizični obliki, je bilo izdelanih kar nekaj različnih računalniških testov, kjer se namesto tapkanja po fizičnih kockah »prižigajo« in »ugašajo« kvadrati na računalniškem zaslonu. Preizkušanec mora v tem primeru ponoviti zaporedje s klikanjem z miško na kvadrate [Pearson and Sahraie, 2003, LeFevre et al., 2010] oziroma s tapkanjem na kvadrate na zaslonu (na dotik) [Smyth and Scholley, 1994, Vandierendonck et al., 2004, Brunetti et al., 2014]. Narejena je bila tudi verzija testa s haptičnim odzivom [Ruggiero and Iachini, 2010], test s hojo (t.i. walking corsi) [Piccardi et al., 2008] in celo s hojo v navidezni resničnosti [Nori et al., 2015]. Pomanjkljivost takšnih testov je predvsem v tem, da se od preizkušanca pričakuje, da mu je »manipulacija« z računalnikom domača in mu to ne povzroča večjih težav in prevelikega stresa. Poleg tega pa mora biti oseba fizično v stanju, ko lahko premika roke oz. hodi. Glede na to, da so v našem primeru ciljna skupina predvsem starostniki, ki v veliki meri niso večji uporabe sodobnih tehnologij ali pa so telesno ovirani, moramo razmisliti o načinu izvedbe testa, ki od njih ne zahteva fizične manipulacije (z rokami, hojo, ipd.) z računalnikom. Kot možna rešitev se pojavijo sistemi

za spremljanje očesnih gibov, ki bi omogočili, da bi preizkušanci izvedli test Corsi zgolj z gledanjem na računalniški zaslon.

3 SISTEMI ZA SPREMLJANJE OČESNIH GIBOV

Sistemi za spremljanje očesnih premikov (*angl. eye-tracker*) so v zadnjih letih tehnično zelo napredovali tako po zmogljivostih kot tudi po cenovni dostopnosti in razširjenosti. Postali so manj občutljivi na premike glave med snemanjem (posledično ni več potrebno uporabniku fiksirati brado, kar je bilo zanj izjemno neudobno), izboljšala se je programska oprema za natančnejši zajem podatkov, povečali sta se natančnost in frekvenca snemanja. Nekateri sistemi so popolnoma integrirani v okvirje klasičnih očal in so s tem postali še bolj priročni za uporabo.

Med najbolj razširjenimi sistemi za sledenje očesnim gibom so tisti, ki delujejo na osnovi videa. Tovrstni sistemi so neinvazivni in uporabljajo relativno cenovno ugodno strojno opremo. Sistemi uporabnikove oči snemajo z več (vsaj dvema) majhnimi kamerami, ki delujejo v infrardečem spektru. Programska oprema pa z uporabo algoritmov za analizo videa iz posnetkov izlušči naslednje podatke: sliko, ki jo vidi (dobi na retino) uporabnik naprave, odziv okulomotorne sistema, ki vključuje odzivni čas in točne karakteristike sakad (*angl. saccades*), pogostost mežikanja in velikost zenic v času. To nam omogoča, da lahko spremljamo očesne gibe sledenja, ki so lahko moteni pri številnih nevroloških obolenjih. Tovrstne naprave torej raziskovalcem omogočajo na objektivni način (preko meritev v realnem času) opazovati in analizirati vzorce očesnih premikov in usmerjanja vidne pozornosti. Kvantifikacija podatkov s svojo objektivnostjo in primerljivostjo (skozi čas) nudi tudi

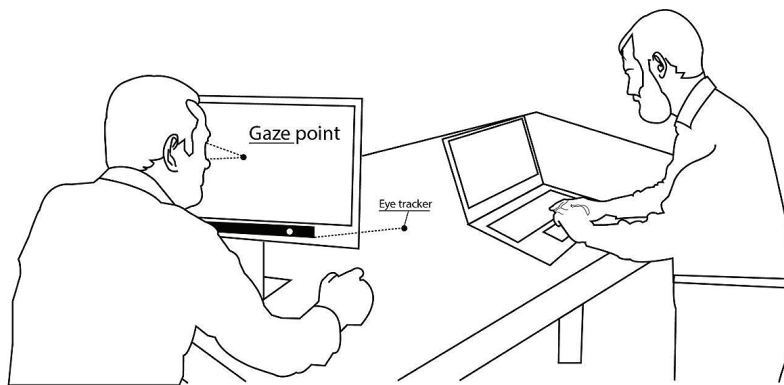
možnost objektivnega spremljanja boleznin skozi čas.

Vse to so razlogi, da so v zadnjih letih tovrstni sistemi postali eno bolj razširjenih orodij pri raziskavah na področju nevroloških obolenj, kot so kognitivni upad [Crutcher et al., 2009, Wilcockson et al., 2019, Tao et al., 2020], parkinsonova bolezen [Blekher et al., 2009, Jansson et al., 2015], motnje avtističnega spektra [Chita-Tegmark, 2016], Rettov sindrom [Rose et al., 2013] in drugi.

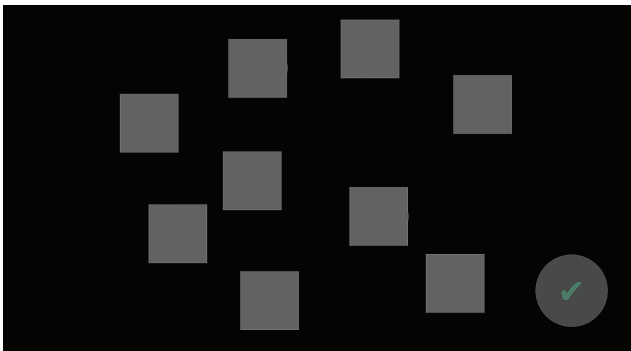
4 DIGITALIZACIJA IN PRILAGODITEV TESTA CORSI ZA UPORABO S SISTEMOM ZA SPREMLJANJE OČESNIH GIBOV

Kolikor nam je znano, do zdaj še ni bilo razvitega testa Corsi, ki bi ga preizkušanec lahko opravljal zgolj s pogledom na računalniški zaslon. Poleg tega si ne želimo, da bi moral preizkušanec na kakršen koli način manipulirati z računalnikom, saj bi mu s tem lahko povzročili dodaten stres, kar bi lahko vodilo do napačnih rezultatov. Zato poleg preizkušanca potrebujemo tudi preizkuševalca, ki poskrbi za zagon programa za izvedbo testa in kalibracijo sistema za sledenje očesnim gibom. Da preizkušanca ne obremenimo z dodatnimi postopki, predlagamo, da imata preizkuševalec in preizkušanec ločene zaslone. Preizkušanec vidi zgolj nalogo, ki jo mora izvesti, preizkuševalec pa lahko nadzira celoten potek izvajanja testiranja, zažene in zaustavi test, opravi kalibracijo, ipd. Na sliki 2 je prikazana shema predlagane postavitve sistema.

Pri digitalizaciji testa Corsi smo se želeli čim bolj približati osnovnemu testu Corsi [Corsi, 1972]. Zaradi tega smo ohranili devet elementov, ki so razporejeni na računalniškem zaslonu. Pri postavitvi smo izhajali iz slike 1, pri čemer smo jo za potrebe pogleda



Slika 2: Postavitev celotnega sistema za izvedbo testiranja. Levo je preizkušanec; njegov pogled spremlja in beleži sistem za spremljanje očesnih gibov. Desno je preizkuševalec, ki na ločenem zaslonu nadzuruje potek testiranja.



Slika 3: Prvotna postavitev elementov testa Corsi na zaslonu.

s strani preizkušanca morali zavrteti za 180 stopinj (na sliki 1 je namreč predstavljen pogled preizkuševalca). Pri prvi implementaciji testa smo poleg postavitve ohranili pravokotno obliko elementov (kock), kot je razvidno iz slike 3. Posamezno zaporedje je bilo prikazano tako, da se je posamezen kvadrat v zaporedju po vrsti za trenutek (150 ms) obarval v zeleno barvo in nato nazaj v sivo. Po enako dolgem premoru (150 ms) se je obarval naslednji kvadrata. Ko je preizkušavec moral ponoviti zaporedje, se je izbrani kvadrat prav tako prižgal zgolj za kratek čas (oz. utripnil), kar je nakazalo označitev kvadrata. Ko je preizkušavec menil, da je označil vse potrebne kvadrate, je izvajanje trenutnega zaporedja zaključil s pogledom na kljukico v desnem spodnjem kotu. Vsaka dolžina zaporedja je imela dve različici, ki sta se izvedli zaporedno. Če je preizkušavec vsaj eno izmed različic posamezne dolžine zaporedja pravilno označil, se je testiranje nadaljevalo z zaporedjem, ki je bilo za en kvadrata daljše. Najdaljše zaporedje je bilo dolžine 9 (kolikor je različnih kvadratkov).

4.1 Testiranje uporabniške izkušnje

Pred vključitvijo digitalizirane verzije testa Corsi v končno baterijo testov za odkrivanje MCI, smo izvedli krajše testiranje uporabniške izkušnje, s katero smo želeli ugotoviti, če je prilagojeni test razumljiv in če je časovna komponenta primerno zasnovana.

Testiranje smo izvedli v kontroliranem okolju, vključili pa smo 24 oseb različnih starostnih in izobrazbenih profilov, ki smo jih izbrali med sodelavci in prostovoljci. Najmlajša oseba je bila stara 30 let, najstarejša pa 79. Vsak preizkušavec je testiranje opravljal dvakrat (označevanje zaporedij od prvega označenega kvadratka do zadnjega in obratno). Med izvajanjem testiranja s posamezno osebo, osta-

le osebe, ki še niso bile testiranje, niso bile prisotne. Preizkuševalca je na svojem zaslonu spremljal, kam preizkušavec v določenem trenutku gleda in si zapisoval svoja opažanja. Podatki iz sistema za sledenje očesnih gibov pa so se avtomatsko shranjevali na računalnik. Preizkušavce smo na koncu povprašali za morebitne komentarje, nekateri izmed njih pa so jih podali že med izvajanjem testa.

Na podlagi testiranja uporabniške izkušnje smo prišli do naslednjih ugotovitev:

- zaporedje je prehitro prikazano (150 ms) in zmede preizkušavca;
Rešitev: podaljšanje časa obarvanja kvadratka in premora med posameznimi obarvanji na 250 ms;
- če pri ponavljanju zaporedja izbrani kvadrata zgolj »utripne«, uporabnika zmede, saj ni prepričan, da je bil kvadrata res izbran;
Rešitev: kvadrata po izboru ostane prižgan;
- čas gledanja kvadratka za izbiro je prekratek (200 ms) in se velikokrat nehote izbere napačen kvadrata, ki ga je preizkušavec s pogledom zgolj prečkal;
Rešitev: podaljšanje časa potrebne fiksacije za izbor kvadratka na 400 ms;
- kvadratki na levi strani so nekoliko preveč skupaj in ob slabši kalibraciji lahko pride to tega, da se označi drug kvadrata, kot ga je preizkušavec dejansko gledal;
Rešitev: namesto kvadratkov se izriše krogec in s tem nekoliko »odmaknemo« elemente med seboj;
- če se testiranec zmoti, ne more popraviti svojega izbora, kljub temu, da ve, da je naredil napako (te možnosti ni imel niti preizkuševalca);
Rešitev: levo spodaj se doda krogec z znakom X - s fiksacijo na ta krogec se razveljavi izbor kvadratkov v trenutnem zaporedju in preizkušavec lahko na novo začne z izborom zaporedja;
- preizkušavec ni bil prepričan ali računalnik pravilno zaznava njegov pogled ali ne in koliko časa mora še gledati posamezen element na zaslonu;
Rešitev: takoj, ko računalnik zazna, da preizkušavec gleda v posamezen element na zaslonu, se pojavi animacija okoli tistega elementa (progres), ki nakazuje, koliko časa že gleda dotičen element in koliko časa še mora gledati tja, da bo element označen kot izbran;
- označevanje zaporedja od zadnjega označenega kvadratka proti prvemu je, pri takšni postavitvi kvadratkov na zaslonu, pretežka;
Rešitev: test naj se izvaja le v eno smer - torej od prvega označenega kvadratka do zadnjega.

Na podlagi teh ugotovitev, smo pripravili novo implementacijo testa Corsi v kateri smo zajeli rešitve, ki so predlagane zgoraj. Takšna rešitev je implementirana v bateriji testov za odkrivanje MCI, ki je v kliničnem preizkušanju. Posnetek implementacije je viden na sliki 6 oz. 7 in opisana v poglavju 4.2.

4.2 Implementacija testa Corsi

Test Corsi smo implementirali za uporabo s sistemom za spremljanje očesnih gibov podjetja Tobii in sicer za model 4C (prikazan na sliki 4), ki je med cenovno bolj dostopnimi modeli, ki še zagotavljajo dovolj dobro natančnost merjenja. Zajem podatkov uporabljenega modela je med 60 in 90 Hz, deluje pa na zaslonih do velikosti diagonale 68,58 cm (27") v razmerju 16:9. Sistem ima na voljo knjižnico SDK, s katero lahko dostopamo do podatkov, ki jih je sistem zajel. Deluje tako na operacijskem sistemu Linux kot tudi na operacijskem sistemu Windows.

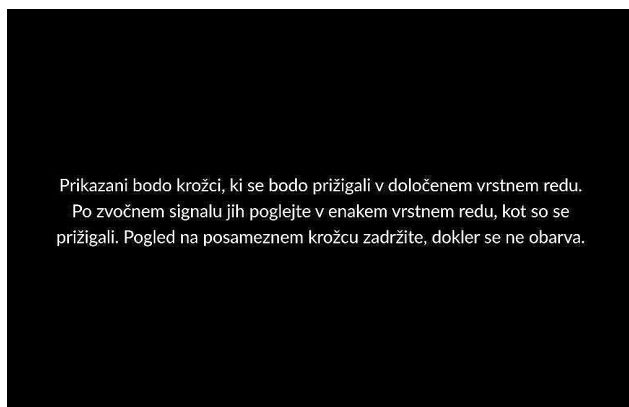


Slika 4: Sistem za spremljanje očesnih gibov Tobii Eye Tracker 4C. (Slika je povzeta s spletne strani www.tobii.com.)

Za naše potrebe smo uporabili zaslon z velikostjo diagonale 60,96 cm (24") in resolucijo 1920 x 1080 px, ki je imel na spodnjem robu nameščen sistem za spremljanje gibanja oči. Preizkuševalec je uporabljal prenosni računalnik z nameščenim operacijskim sistemom Debian.

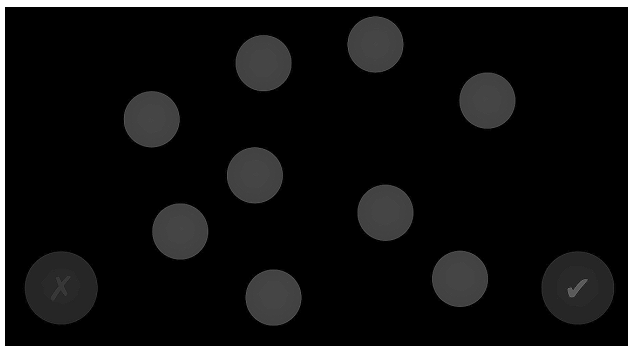
Pred začetkom testiranja se preizkušancu na zaslonu izpišejo kratka navodila (slika 5), ki so hkrati predvajana tudi v zvočni obliki.

Navodilom sledi test. Na zaslonu se prikaže devet krožcev (slika 6), ki so enake barve. Začetek prikaza posameznega zaporedja s prižiganjem krožcev je napovedan s kratkim zvočnim signalom (pisk), da oseba opozori, da mora biti pozorna na dogajanje na zaslonu. Prižiganje krožcev se prične 1000 ms ko pisku.



Slika 5: Zaslon s kratkimi navodili za izvajanje testa Corsi.

T.i. prižiganje krožcev pomeni, da se posamezen krožec za hip (250 ms) obarva zeleno in nato »uga-sne«. Naslednji krožec se obarva 250 ms po prejšnjem.

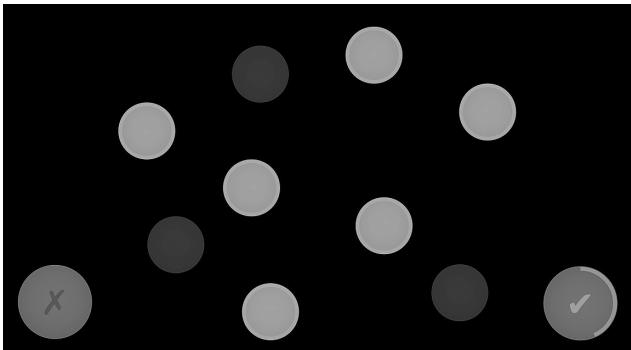


Slika 6: Zaslon z razporeditvijo devetih točk testa Corsi.

Prikazu zaporedja ponovno sledi kratek zvočni signal (pisk), ki preizkušanca opozori, da je na vrsti, da ponovi zaporedje. Posamezen krožec mora gledati toliko časa (fiksacija), dokler se ne obarva zeleno. Ko preizkušane meni, da je pravilno ponovil zaporedje, mora pogled usmeriti v desni spodnji kot, kjer je krožec z zeleno kljukico. Tega mora gledati 2 s (da se izognemo napakam, do katerih bi lahko prišlo, če bi oseba po nesreči s pogledom oplazila za krožec) in s tem se zaključi testiranje trenutnega zaporedja. Primer označitve zaporedja s pogledom je prikazan na sliki 7.

Če se preizkušanec med izvajanjem naloge zmoti, lahko napako popravi s pogledom v levi spodnji kot (kjer je v krožcu rdeč križec). Tudi na tem krožcu mora pogled zadržati dlje časa (2s), s tem pa razveljavi celotno zaporedje, ki ga je označil s pogledom in prične z označevanjem od začetka.

Podobno kot v izvornem testu Corsi [Corsi, 1972], se posamezna dolžina zaporedja obarvanih krožcev ponovi dvakrat. Izjema je najkrajše zaporedje, ki je dolžine dve, saj se ponovi trikrat. Razlog je v tem, da se preizkušanci na začetku še nekoliko lovijo in se lahko zgodi, da po nepotrebnem naredijo kakšno napako. Test se izvaja tako dolgo, dokler preizkušanec pri posamezni dolžini zaporedja vsaj eno izmed zaporedij označi v pravilnem zaporedju. Najdaljše zaporedje je torej dolžine devet, kolikor je krožcev na zaslonu. Testiranje se predčasno zaključí, ko preizkušanec napačno ponovi vse različice iste dolžine zaporedja. Z nadaljevanjem testiranja bi namreč osebo po nepotrebnem dodatno obremenjevali in jim povzročali psihični stres, ker nečesa niso zmožni ponoviti.



Slika 7: Zaslon z označenimi krožci, kot jih je označil preizkušanec.

5 SKLEP IN NADALJNJE DELO

V članku je predstavljen predlog digitalizacije testa Corsi za uporabo s sistemom za sledenje očesnih gibov. Prednost digitalizirane verzije testa v primerjavi s klasično je predvsem v možnosti avtomatske analize podatkov in postavitvi ocene kratkoročnega vidno-prostorskega spomina preizkušanca, razbremenitvi psihologov - saj je za izvedbo testa potrebna zgolj prisotnost tehnika, ki zažene kalibracijo in test, služi pa lahko tudi kot hiter (presejalni) test na primarni ravni, na podlagi katerega se lahko osebni zdravnik odloči o nadaljnji napotitvi preizkušanca h specialistu.

Pri digitalizaciji tovrstnih testov se je treba zavedati, komu so testi namenjeni. Ker gre v našem primeru za starejše osebe, ki niso nujno večje uporabe modernih tehnologij IKT, poleg tega pa so morda tudi telesno ovrane, je potrebno test čim bolj poenostaviti za uporabo. Odstraniti je treba vse nepotrebne informacije in manipulacije, ki bi lahko pritegnile pozornost preizkušanca in s tem privedle do napačnih rezultatov testiranja. Ker gre v našem primeru za uporabo digitaliziranega testa skupaj s sistemom za spremljanje očesnih gibov, je pred izvedbo testa potrebno narediti čim bolj natančno kalibracijo, saj smo le tako lahko prepričani, da naš sistem pravilno beleži pogled preizkušanca. Za označitev, da je preizkušanec pogledal določen krog na zaslonu, je potrebno določiti, koliko časa mora posamezen krog gledati. Če je čas prekratek, lahko napačno označimo krog, ki ga je preizkušanec nehote ošvrknil s pogledom. Če je predolg, lahko zaradi mežikanja izgubimo »signal« pogleda ali pa oseba pozabi, kateri je krog, ki bi moral slediti trenutnemu krogu.

Predlagana digitalizirana verzija testa Corsi je že implementirana in je del baterije testov, ki se trenutno uporablja kot del raziskave v klinični praksi. Na podlagi podatkov, ki jih bomo dobili, bomo z uporabo algoritmov umetne inteligence zgradili diagnostični model, s katerim bomo lahko avtomatsko napovedovali oceno stanja kratkoročnega vidno-prostorskega spomina preizkušanca in, če bo mogoče, napovedali, ali ima oseba prisotne znake blagega kognitivnega upada ali ne.

LITERATURA

- [1] Alzheimer's Disease International, L. (2019). *World Alzheimer Report 2019: Attitudes to dementia*. Alzheimer's Disease International (ADI).
- [2] Ayati, Z., Chang, D., and Lake, J. (2020). Advances in treatment of mild cognitive impairment (mci) and dementia: A review of promising non-pharmaceutical modalities. *Frontiers in Clinical Drug Research-Dementia: Volume 1*, 1:78.
- [3] Benedict, C., Byberg, L., Cedernaes, J., Hogenkamp, P. S., Giedratís, V., Kilander, L., Lind, L., Lannfelt, L., and Schiöth, H. B. (2015). Self-reported sleep disturbance is associated with alzheimer's disease risk in men. *Alzheimer's & Dementia*, 11(9):1090-1097.
- [4] Berch, D. B., Krikorian, R., and Huha, E. M. (1998). The corsi block-tapping task: Methodological and theoretical considerations. *Brain and cognition*, 38(3):317-338.
- [5] Blekher, T., Weaver, M., Rupp, J., Nichols, W. C., Hui, S. L., Gray, J., Yee, R. D., Wojcieszek, J., and Foroud, T. (2009). Multiple step pattern as a biomarker in parkinson disease. *Parkinsonism & related disorders*, 15(7):506-510.
- [6] Borson, S., Scanlan, J., Brush, M., Vitaliano, P., and Dokmak, A. (2000). The mini-cog: a cognitive 'vital signs' measure for dementia screening in multi-lingual elderly. *International journal of geriatric psychiatry*, 15(11):1021-1027.

- [7] Brunetti, R., Del Gatto, C., and Delogu, F. (2014). ecorsi: implementation and testing of the corsi block-tapping task for digital tablets. *Frontiers in Psychology*, 5:939.
- [8] Cass, S. P. (2017). Alzheimer's disease and exercise: a literature review. *Current sports medicine reports*, 16(1):19–22.
- [9] Chang, W.-P., Liu, M.-E., Chang, W.-C., Yang, A. C., Ku, Y.-C., Pai, J.-T., Huang, H.-L., and Tsai, S.-J. (2013). Sleep apnea and the risk of dementia: a population-based 5-year follow-up study in taiwan. *PLoS one*, 8(10):e78655.
- [10] Chita-Tegmark, M. (2016). Social attention in asd: a review and meta-analysis of eye-tracking studies. *Research in developmental disabilities*, 48:79–93.
- [11] Corsi, P. M. (1972). Human memory and the medial temporal region of the brain. *doctoral dissertation, McGill University*.
- [12] Crooks, V. C., Lubben, J., Petitti, D. B., Little, D., and Chiu, V. (2008). Social network, cognitive function, and dementia incidence among elderly women. *American journal of public health*, 98(7):1221–1227.
- [13] Crutcher, M. D., Calhoun-Haney, R., Manzanares, C. M., Lah, J. J., Levey, A. I., and Zola, S. M. (2009). Eye tracking during a visual paired comparison task as a predictor of early dementia. *American Journal of Alzheimer's Disease & Other Dementias*, 24(3):258–266.
- [14] de Paula, J. J., Moreira, L., Nicolato, R., De Marco, L. A., Côrea, H., Romano-Silva, M. A., De Moraes, E. N., Bicalho, M. A., and Malloy-Diniz, L. F. (2012). The tower of london test: Different scoring criteria for diagnosing alzheimer's disease and mild cognitive impairment. *Psychological Reports*, 110(2):477–488.
- [15] Féart, C., Samieri, C., Rondeau, V., Amieva, H., Portet, F., Dartigues, J.-F., Scarmeas, N., and Barberger-Gateau, P. (2009). Adherence to a mediterranean diet, cognitive decline, and risk of dementia. *Jama*, 302(6):638–648.
- [16] Firth, J., Stubbs, B., Vancampfort, D., Schuch, F., Lagopoulos, J., Rosenbaum, S., and Ward, P. B. (2018). Effect of aerobic exercise on hippocampal volume in humans: a systematic review and meta-analysis. *Neuroimage*, 166:230–238.
- [17] Folstein, M. F., Folstein, S. E., and McHugh, P. R. (1975). »mini-mental state«: a practical method for grading the cognitive state of patients for the clinician. *Journal of psychiatric research*, 12(3):189–198.
- [18] Fratiglioni, L., Paillard-Borg, S., and Winblad, B. (2004). An active and socially integrated lifestyle in late life might protect against dementia. *The Lancet Neurology*, 3(6):343–353.
- [19] Granda, G., Mlakar, J., and Vodušek, D. B. (2003). Kratek preizkus spoznavnih sposobnosti-umerjanje pri preiskovancih, starih od 55 do 75 let (i). *Slovenian Medical Journal*, 72(10).
- [20] Grant, W. B. (1999). Dietary links to alzheimer's disease: 1999 update. *Journal of Alzheimer's Disease*, 1(4-5):197–201.
- [21] Grant, W. B. (2016). Using multicountry ecological and observational studies to determine dietary risk factors for alzheimer's disease. *Journal of the American College of Nutrition*, 35(5):476–489.
- [22] Guariglia, C. C. (2007). Spatial working memory in alzheimer's disease: a study using the corsi block-tapping test. *Dementia & Neuropsychologia*, 1(4):392–395.
- [23] Hamer, M. and Chida, Y. (2009). Physical activity and risk of neurodegenerative disease: a systematic review of prospective evidence. *Psychological medicine*, 39(1):3–11.
- [24] Hugo, J. and Ganguli, M. (2014). Dementia and cognitive impairment: epidemiology, diagnosis, and treatment. *Clinics in geriatric medicine*, 30(3):421–442.
- [25] Jansson, D., Medvedev, A., Axelson, H., and Nyholm, D. (2015). Stochastic anomaly detection in eye-tracking data for quantification of motor symptoms in parkinson's disease. In *Signal and Image Analysis for Biomedical and Life Sciences*, pages 63–82. Springer.
- [26] Kaplan, E. (1991). *WAIS-R as a Neuropsychological Instrument (WAIS-R NI)*. Psychological Corporation.
- [27] Kessels, R. P., Van Zandvoort, M. J., Postma, A., Kappelle, L. J., and De Haan, E. H. (2000). The corsi block-tapping task: standardization and normative data. *Applied neuropsychology*, 7(4):252–258.
- [28] LeFevre, J.-A., Fast, L., Skwarchuk, S.-L., Smith-Chant, B. L., Bisanz, J., Kamawar, D., and Penner-Wilger, M. (2010). Pathways to mathematics: Longitudinal predictors of performance. *Child development*, 81(6):1753–1767.
- [29] Letenneur, L. (2004). Risk of dementia and alcohol and wine consumption: a review of recent results. *Biological research*, 37(2):189–193.
- [30] Lim, A. S., Kowgier, M., Yu, L., Buchman, A. S., and Bennett, D. A. (2013). Sleep fragmentation and the risk of incident alzheimer's disease and cognitive decline in older persons. *Sleep*, 36(7):1027–1032.
- [31] Lourida, I., Hannon, E., Littlejohns, T. J., Langa, K. M., Hyppönen, E., Kuz'ma, E., and Llewellyn, D. J. (2019). Association of lifestyle and genetic risk with incidence of dementia. *Jama*, 322(5):430–437.
- [32] Lourida, I., Soni, M., Thompson-Coon, J., Purandare, N., Lang, I. A., Ukoumunne, O. C., and Llewellyn, D. J. (2013). Mediterranean diet, cognitive function, and dementia: a systematic review. *Epidemiology*, pages 479–489.
- [33] McColgan, P., Evans, J. R., Breen, D. P., Mason, S. L., Barker, R. A., and Williams- Gray, C. H. (2012). Addenbrooke's cognitive examination-revised for mild cognitive impairment in parkinson's disease. *Movement Disorders*, 27(9):1173–1177.
- [34] Mioshi, E., Dawson, K., Mitchell, J., Arnold, R., and Hodges, J. R. (2006). The addenbrooke's cognitive examination revised (ace-r): a brief cognitive test battery for dementia screening. *International Journal of Geriatric Psychiatry: A journal of the psychiatry of late life and allied sciences*, 21(11):1078–1085.
- [35] Nasreddine, Z. S., Phillips, N. A., Bédirian, V., Charbonneau, S., Whitehead, V., Collin, I., Cummings, J. L., and Chertkow, H. (2005). The montreal cognitive assessment, moca: a brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699.
- [36] Nori, R., Piccardi, L., Migliori, M., Guidazzoli, A., Frasca, F., De Luca, D., and Giusberti, F. (2015). The virtual reality walking corsi test. *Computers in Human Behavior*, 48:72–77.
- [37] Parsey, C. M. and Schmitter-Edgecombe, M. (2011). Quantitative and qualitative analyses of the clock drawing test in mild cognitive impairment and alzheimer disease: evaluation of a modified scoring system. *Journal of geriatric psychiatry and neurology*, 24(2):108–118.
- [38] Pearson, D. and Sahraie, A. (2003). Oculomotor control and the maintenance of spatially and temporally distributed events in visuo-spatial working memory. *The Quarterly Journal of Experimental Psychology Section A*, 56(7):1089–1111.
- [39] Piccardi, L., Iaria, G., Ricci, M., Bianchini, F., Zompanti, L., and Guariglia, C. (2008). Walking in the corsi test: which type of memory do you need? *Neuroscience Letters*, 432(2):127–131.
- [40] Rainville, C., Lepage, E., Gauthier, S., Kergoat, M.-J., and Belleville, S. (2012). Executive function deficits in persons with mild cognitive impairment: a study with a tower of london task. *Journal of Clinical and Experimental Neuropsychology*, 34(3):306–324.

- [41] Rakuša, M., Granda, G., Kogoj, A., Mlakar, J., and Vodušek, D. (2006). Mini-mental state examination: standardization and validation for the elderly slovenian population. *European journal of neurology*, 13(2):141–145.
- [42] Rose, S. A., Djukic, A., Jankowski, J. J., Feldman, J. F., Fishman, I., and Valicenti-Mcdermott, M. (2013). Rett syndrome: an eye-tracking study of attention and recognition memory. *Developmental Medicine & Child Neurology*, 55(4):364–371.
- [43] Ruggiero, G. and Iachini, T. (2010). The role of vision in the corsi block-tapping task: Evidence from blind and sighted people. *Neuropsychology*, 24(5):674.
- [44] Smyth, M. and Scholey, K. (1994). Interference in spatial immediate memory. *Memory and Cognition*, 22:1–13.
- [45] Solfrizzi, V., Panza, F., Frisardi, V., Seripa, D., Logroscino, G., Imbimbo, B. P., and Pilotto, A. (2011). Diet and alzheimer's disease risk factors or prevention: the current evidence. *Expert review of neurotherapeutics*, 11(5):677–708.
- [46] Swiger, K. J., Manalac, R. J., Blumenthal, R. S., Blaha, M. J., and Martin, S. S. (2013). Statins and cognition: a systematic review and meta-analysis of short-and long-term cognitive effects. In *Mayo clinic proceedings*, volume 88, pages 1213–1221. Elsevier.
- [47] Szekely, C. A., Breitner, J. C., Fitzpatrick, A. L., Rea, T. D., Psaty, B. M., Kuller, L. H., and Zandi, P. P. (2008). Nsaid use and dementia risk in the cardiovascular health study*: Role of apoe and nsaid type. *Neurology*, 70(1):17–24.
- [48] Tao, L., Wang, Q., Liu, D., Wang, J., Zhu, Z., and Feng, L. (2020). Eye tracking metrics to screen and assess cognitive impairment in patients with neurological disorders. *Neurological Sciences*, pages 1–8.
- [49] Vandierendonck, A., Kemps, E., Fastame, M. C., and Szmalec, A. (2004). Working memory components of the corsi blocks task. *British journal of psychology*, 95(1):57–79.
- [50] Wang, H.-X., Karp, A., Winblad, B., and Fratiglioni, L. (2002). Late-life engagement in social and leisure activities is associated with a decreased risk of dementia: a longitudinal study from the kungsholmen project. *American journal of epidemiology*, 155(12):1081–1087.
- [51] Wilcockson, T. D., Mardanbegi, D., Xia, B., Taylor, S., Sawyer, P., Gellersen, H. W., Leroi, I., Killick, R., and Crawford, T. J. (2019). Abnormalities of saccadic eye movements in dementia due to alzheimer's disease and mild cognitive impairment. *Aging (Albany NY)*, 11(15):5389.
- [52] Wong, W. B., Lin, V. W., Boudreau, D., and Devine, E. B. (2013). Statins in the prevention of dementia and alzheimer's disease: a meta-analysis of observational studies and an assessment of confounding. *Pharmacoepidemiology and drug safety*, 22(4):345–358.
- [53] Yip, A. G., Green, R. C., Huyck, M., Cupples, L. A., and Farrer, L. A. (2005). Nonsteroidal anti-inflammatory drug use and alzheimer's disease risk: the mirage study. *BMC geriatrics*, 5(1):1–6.

■

Vida Groznik je asistentka za računalništvo na Fakulteti za matematiko, naravoslovje in informacijske tehnologije na Univerzi na Primorskem ter raziskovalka v Laboratoriju za umetno inteligenco na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Doktorirala je leta 2018 z delom, ki je tesno povezovalo umetno inteligenco in nevrologijo. Je (so)avtorica večih raziskovalnih člankov in poglavij v knjigah s področja umetne inteligence v medicini. Pridobila in delala je na večih projektih financiranih s programov EU, Ministrstva za izobraževanje, znanost in šport ter Slovenskega podjetniškega sklada. Je soustanoviteljica in direktorica podjetja NEUS Diagnostics, d.o.o. v okviru katerega razvijajo sisteme za pomoč pri oceni zdravstvenega stanja uporabnikov.

Iz Islovarja

Islovar je spletni terminološki slovar informatike, ki ga objavlja jezikovna sekcija Slovenskega društva INFORMATIKA in ga najdete na naslovu <http://www.islovar.org>. Vabimo vas, da tudi vi prispevate svoje pripombe, predloge ali nove izraze.

blokovni algoritem -ega -tma m (*angl. block algorithm*) simetrični šifrirni algoritem, s katerim se naenkrat šifrira celoten blok; prim. pretočni algoritem

blokovno šifriranje -ega -a s (*angl. block encryption, block cipher mode*) simetrično šifriranje, pri katerem se uporablja blokovni algoritem; prim. pretočno šifriranje

dekodiranje -a s (*angl. decoding*) pretvarjanje kodiranega sporočila z uporabo dogovorjene kode v prvotno obliko; prim. kodiranje

dešifriranje -a s (*angl. decryption, decyphermnt, decrypting, decipherment*) postopek, pri katerem se tajnopis z uporabo šifrirnega algoritma in šifrirnega ključa spremeni v čistopis; prim. šifrirati

kóda -e ž (*angl. code*) sistem pravil za pretvorbo podatkov v drugo obliko, včasih skrajšano ali tajno, za komunikacijo ali shranjevanje; prim. šifra, psevdokoda, strojna koda

kodiranje -a s (*angl. encoding*) pretvarjanje sporočila v drugačno zaporedje znakov z uporabo dogovorjene kode; prim. dekodiranje

kodiranje signála -a -- s (*angl. scrambling*) spreminjanje signala s podatki tako, da ga prejemnik prepozna samo s podobno napravo

oznaka -e ž (*angl. identifier, label, ID*) kar je mogoče uporabiti za enolično označevanje česa; sin. identifikator, šifra¹

pretočni algoritem -ega -tma m (*angl. stream algorithm*) simetrični šifrirni algoritem, s katerim se šifrira tok podatkov; prim. blokovni algoritem

pretočno šifriranje -ega -a s (*angl. stream encryption*) simetrično šifriranje, pri katerem se uporablja pretočni algoritem; prim. blokovno šifriranje

programska kóda -e -e ž (*angl. program code*) zapis računalniškega programa ali dela računalniškega programa v enem od programskih jezikov; sin. koda

simétrični šifrirni algorítem -ega -ega -tma m (*angl. symmetric encryption algorithm*) šifrirni algoritem, ki za šifriranje in dešifriranje uporablja isti tajni ključ; prim. asimetrični šifrirni algoritem

šifra¹ -e ž (*angl. identifier, label*) kar je mogoče uporabiti za enolično označevanje česa; sin. oznaka, identifikator

šifra² -e ž (*angl. code, cipher*) pravilo za pretvorbo sporočila v neprepoznavno obliko; prim. koda, šifrirni ključ

šifra³ -e ž (*angl. password*) gl. geslo

šifránt -a m (*angl. code list*)

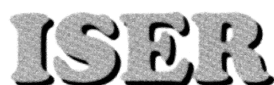
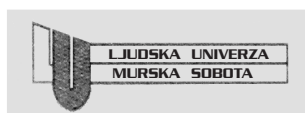
1. urejen seznam šifer¹ z razlago pomena
2. nabor kriptografskih kod

šifíranje -a s (*angl. encryption, encipherment*) postopek, pri katerem se z uporabo šifrirnega algoritma in šifrirnega ključa čistopis spremeni v tajnopis; prim. dešifriranje

šifírní kljúč -ega -a m (*angl. key, encryption key*) niz znakov, ki služi za kodiranje in dekodiranje podatkov; prim. šifra

Izpitni centri ECDL

ECDL (European Computer Driving License), ki ga v Sloveniji imenujemo evropsko računalniško spričevalo, je standardni program usposabljanja uporabnikov, ki da zaposlenim potrebno znanje za delo s standardnimi računalniškimi programi na informatiziranem delovnem mestu, delodajalcem pa pomeni dokazilo o usposobljenosti. V Evropi je za uvajanje, usposabljanje in nadzor izvajanja ECDL pooblaščen ustanova ECDL Foundation, v Sloveniji pa je kot član CEPIS (Council of European Professional Informatics) to pravico pridobilo Slovensko društvo INFORMATIKA. V državah Evropske unije so pri uvajanju ECDL močno angažirane srednje in visoke šole, aktivni pa so tudi različni vladni resorji. Posebno pomembno je, da velja spričevalo v 148 državah, ki so vključene v program ECDL. Doslej je bilo v svetu izdanih že več kot 11,6 milijona indeksov, v Sloveniji več kot 17.000, in podeljenih več kot 11.000 spričeval. Za izpitne centre v Sloveniji je usposobljenih osem organizacij, katerih logotipe objavljamo.



Strokovni prispevki

Bor Krizmanič, Aleš Groznik:
AKTUALNE DEJAVNOSTI, PRIMERNE ZA UPORABO VERIŽENJA
PODATKOVNIH BLOKOV

Mitja Gradišnik, Tina Beranič, Sašo Karakatič:
Z UMETNO INTELIGENCO PODPRT RAZVOJ PROGRAMSKE OPREME

Mateja Kocbek Bule:
ZNANJSKI DELAVCI IN NOTACIJA CMMN

Kratki znanstveni prispevki

Amra Omanović, Polona Oblak, Tomaž Curk:
APPLICATION OF TROPICAL SEMIRING FOR MATRIX FACTORIZATION

Matej Vitek, Peter Rot, Vitomir Štruc, Peter Peer:
POGLOBLJEN POGLED V BELOČNIČNO BIOMETRIJO: NOVA PODATKOVNA
MNOŽICA IN ŠTUDIJA DELOVANJ

Vida Groznik:
DIGITALIZACIJA IN PRILAGODITEV PSIHOLOŠKEGA TESTA ZA UPORABO
S SISTEMOM ZA SPREMLJANJE OČESNIH GIBOV

Informacije

IZ ISLOVARJA

