



04 UPORABNA INFORMATIKA

2022 ◀ ŠTEVILKA 4 ◀ LETNIK XXX ◀ ISSN 1318-1882

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

2022 ŠTEVILKA 4 OKT/NOV/DEC LETNIK XXX ISSN 1318-1882

▣ Znanstveni prispevki

Patrik Rek, Tina Beranič, Marjan Heričko

Študija pričakovanj in uporabnosti orodij za razvoj aplikacij z malo ali nič programiranja

227

▣ Kratki znanstveni prispevki

Davor Hafnar, Jure Demšar

Ena igra za vse: Generiranje personalizirane vsebine v mobilnih igrah

241

Tim Oblak, Peter Peer

Ocenjevanje kakovosti prstnih sledi z ansambli globokega učenja

245

Andraž Krašovec, Veljko Pejovič

Vpliv kognitivne obremenjenosti na vedenjske vzorce posameznikov pri overjanju uporabnikov

250

▣ Strokovni prispevki

Slavko Žitnik, Karmen Kern Pipan, Miha Jesenko, Dejan Lavbič

Ponovno uporabljive semantične spletne komponente: Primer uporabe za interoperabilnost digitalnih javnih storitev

256

▣ Informacije

Iz Islovarja

270

Vabilo na Mednarodni simpozij s področja operacijskih raziskav

271

Ustanovitelj in izdajatelj

Slovensko društvo INFORMATIKA
Litostrojska cesta 54, 1000 Ljubljana

Predstavniki

Niko Schlamberger

Odgovorni urednik

Mirjana Kljajić Borštnar

Uredniški odbor

Andrej Kovačič, Evelin Krmac, 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 Brinkemper, Slavko Žitnik, Tatjana Welzer Družovec, Vesna Bosilj-Vukšić, Vida Groznik, Vladislav Rajkovič

Recenzentski odbor

Aleksander Sadikov, Alenka Baggia, Alena Brezavšček, Alenka Kavčič, Aleš Groznik, Aljaž Košmerlj, Andrej Kovačič, Anton Manfreda, Blaž Rodič, Bor Plestenjak, Borut Batagelj, Borut Werber, Borut Žalik, Boštjan Šumak, Božidar Potočnik, Branko Kavšek, Branko Šter, Ciril Bohak, Danijel Skočaj, Damjan Fujs, David Jelenc, Dejan Georgiev, Dejan Lavbič, Denis Trček, Domen Mongus, Eva Jereb, Eva Krhač, Evelin Krmac, Inna Novalijska, Irena Nančovska Šerbec, Ivan Gerlič, Jernej Vičič, Jure Žabkar, Jurij Mihelič, Katarina Puc, Lovro Šubelj, Luka Čehovin, Luka Pavlič, Marina Trkman, Marjan Heričko, Marjan Krisper, Marjeta Marolt, Marko Bajec, Marko Hölbl, Marko Robnik Šikonja, Matej Klemen, Matevž Pesek, Matjaž Divja, Mirjana Kljajić Borštnar, Mladen Borovič, Muhamed Turkanović, Niko Schlamberger, Nikola Ljubešić, Patricio Bulić, Peter Trkman, Polona Rus, Robert Leskovic, Sandi Gec, Saša Divjak, Simon Vrhovec, Slavko Žitnik, Uroš Godnov, Uroš Rajkovič, Vida Groznik, Vladislav Rajkovič, Vlado Stankovski, Živa Rant

Tehnični urednik

Slavko Žitnik

Lektoriranje angleških izlečkov

Marvelingua (angl.)

Oblikovanje

KOFEIN DIZAJN, d. o. o.

Prelom in tisk

Boex DTP, d. o. o., Ljubljana

Naklada

100 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 prispevke domačih in tujih avtorjev z najširšega področja informatike, ki se nanašajo tako na poslovanju podjetij, javno upravo, družbo in posameznika. Prispevki so lahko znanstvene, strokovne ali informativne narave, še posebno spodbujamo objavo interdisciplinarnih prispevkov. 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, ki so objavljena na naslovu <http://www.uporabna-informatika.si>.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Prispevki 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 prispevek ponovno prejmejo v pregled. Sprejeti prispevki so pred izidom revije objavljeni na spletni strani revije (predobjava), še prej pa končno verzijo prispevka avtorji dobijo v pregled in potrditev. Uredništvo lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če prispevek ne ustreza kriterijem za objavo v reviji.

Pred objavo prispevka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost prispevka in dovoljuje prenos materialnih avtorskih pravic. 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 pomagali 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 prispevek naj obsega največ 40.000 znakov, kratki znanstveni prispevek do 10.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Prispevek naj bo 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 presledek, pri odstavkih ne uporabljajte zamika.

Naslovu prispevka naj sledi polno ime vsakega avtorja, 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 prispevka. Sledi naj prevod naslova povzetka in ključnih besed v angleškem jeziku. V primeru, da oddajate prispevek v angleškem jeziku, velja obratno. Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Na vsako sliko in tabelo se morate v besedilu prispevka sklicevati in jo pojasniti. Če v prispevku 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. Slikam zaslonov se v prispevku izogibajte, 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 IEEE navajanja bibliografskih referenc, v besedilu to pomeni zaporedna številka navajenega vira v oglatem oklepaju (npr. [1]). Na koncu prispevka navedite samo v prispevku uporabljeno literaturo in vire v enotnem seznamu, urejeno po zaporedni številki vira, prav tako v skladu s pravili IEEE. Več o sistemu IEEE, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani https://owl.purdue.edu/owl/research_and_citation/ieee_style/ieee_general_format.html.

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

Študija pričakovanj in uporabnosti orodij za razvoj aplikacij z malo ali nič programiranja

Patrik Rek, Tina Beranič, Marjan Heričko

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor

patrik.rek@um.si, tina.beranic@um.si, marjan.hericko@um.si

Izvleček

Pristop k razvoju aplikacij, ki temelji na uporabi orodij z malo ali nič programiranja, naslavlja že desetletja znane in prisotne izzive, povezane s produktivnostjo in agilnostjo v oddelkih za informatiko. Hkrati omogoča razvoj poslovnih aplikacij tudi strokovnjakom, ki nimajo programerskih znanj, zato predstavlja idealno rešitev za hiter razvoj inovativnih rešitev v procesih digitalne preobrazbe ter uveljavitev poslovno vodenega razvoja. V prispevku predstavljeni rezultati raziskave, v katero so bili vključeni predstavniki slovenskih podjetij ter študenti informatike, zagotavljajo vpogled v stopnjo poznavanja, sprejetosti in uporabnosti orodij za razvoj aplikacij z nič ali malo programiranja v Sloveniji ter nudijo možnost primerjave s svetovnimi trendi. Razkrivajo tudi nekatere razloge, zakaj pristop v našem prostoru še ni širše uporabljen, vsekakor pa velja izpostaviti pozitivna pričakovanja, ki sovpadajo z, v obstoječi literaturi identificiranimi pozitivnimi lastnostmi pristopa in orodij za razvoj aplikacij z malo ali nič programiranja. Z uporabo uveljavljenih modelov kot so SUS (angl. system usability scale), UMUX (angl. usability metric for user experience) in TAM (angl. technology acceptance model) smo ovrednotili uporabnost predstavnika tovrstnih orodij ter tako potrdili njihovo zrelost in primernost za širšo vpeljavo.

Ključne besede: brez kode, malo kode, razvoj, sprejetost, uporabnost

EXPECTATIONS AND USABILITY STUDY OF THE LOW-CODE/NO-CODE APPLICATION DEVELOPMENT TOOLS

Abstract

The application development approach, based on the use of tools with zero or low code, addresses decades-old known and current challenges related to productivity and agility in IT departments. At the same time, it enables the development of business applications even for professionals who do not possess in-depth programming skills, making it an ideal solution for the rapid development of innovative solutions in digital transformation processes and the implementation of business-driven development. The results of the research, which included representatives of Slovenian companies and IT students, provide insight into the level of knowledge, acceptance and usability of tools for developing applications with zero or low code in Slovenia, and furthermore, offer an opportunity for comparison with global trends. They also reveal certain reasons why the approach is not yet widely used in our area, though it is worth highlighting the positive expectations that coincide with the positive features of the approach identified in the existing literature and tools for developing applications with low or no code. Using established models such as SUS (system usability scale), UMUX (usability metric for user experience) and TAM (technology acceptance model), we evaluated the usability of the selected representative of these tools and thus confirmed their maturity and suitability for wider implementation.

Keywords: No-code, low-code, development, acceptance, usability

1 UVOD

Uspešna uporaba in vpeljavo sodobnih informacijskih tehnologij in rešitev omogoča optimizacijo poslovnih

procesov, hkrati pa pomembno prispeva k razvoju novih poslovnih modelov in inovativnih storitev. Zaradi pomanjkanja strokovnjakov s področja infor-

matike ter njihove preobremenjenosti z operativnimi nalogami so se že v preteklosti skušale uveljaviti iniciative, v sklopu katerih so posamezne informacijske rešitve razvijali kar poslovni uporabniki sami, kar je vodilo do t.i. informatike v senci (angl. shadow IT). Z uveljavitvijo koncepta virtualizacije, računalništva v oblaku, tehnologije zabojnikov, predvsem pa neprekinjenega razvoja in mikrostoritvenih arhitektur, pa smo priča vse večji ponudbi in razširjenosti orodij, ki omogočajo hiter razvoj aplikacij tako strokovnjakom s programerskimi znanji, kot tistim, ki teh znanj sicer nimajo, posedujejo pa bogata domenska znanja. Razvoj aplikacij z malo ali nič programiranja (angl. low-code/no-code application development) je tesno povezan s pristopom, ki k razvoju aplikacij spodbuja tudi zaposlene, ki sicer niso iz področja informatike. Pristop razvoja aplikacij brez poglobljenih znanj programiranja, v angleščini imenovan »citizen development«, tako naslavlja aktualne izzive na področju razvoja informacijskih rešitev, t.j. vse večje potrebe po informacijski podpori, inovativnosti in agilnosti.

1.1 Značilnosti razvoja aplikacij z malo ali nič programiranja

Razvoj z malo ali nič programiranja je pristop, čigar začetki segajo v osemdeseta leta, ko so bila predstavljena orodja za hiter razvoj aplikacij (angl. rapid application development). Ta orodja so služila kot alternativa tradicionalnim razvojnim okoljem, zasnovanim na pisanju kode (Vincent, Driver, & Wong, 2019). Platforme za razvoj aplikacij z malo ali nič programiranja se uporabljajo za hiter razvoj in uvajanje aplikacij po meri z zmanjšanjem ročnega pisanja kode kolikor je mogoče. Takšne platforme, ki jih označujemo z akronimom LCNC (angl. low-code/no-code), omogočajo razvoj celotnih poslovnih aplikacij, vključno z uporabniškim vmesnikom, poslovno logiko, de-

lovnimi tokovi in podatkovnimi storitvami (Vincent et al., 2019). So platforme, ki omogočajo razvoj aplikacij s pomočjo konceptov vizualnega programiranja po metodi povleci in spusti. Gradnja aplikacij torej namesto na pisanju programske kode temelji na grafičnem uporabniškem vmesniku z možnostjo kompozicije gradnikov in storitev ter preprostim spreminjanjem njihovih lastnosti. S tem omogočajo avtorjem aplikacij, da se osredotočijo predvsem na poslovno logiko in potek poslovnih procesov (Hakimi, 2019).

Platforme za razvoj aplikacij z malo ali nič programiranja skrajšajo čas razvoja in zmanjšajo stroške, omogočajo pa tudi preprosto posodabljanje in uvajanje sprememb. Za njihovo uporabo in razvoj aplikacij načeloma ne potrebujemo programerja, saj se lahko razvoja lotijo tudi poslovni uporabniki. Posledično tovrstne platforme zmanjšajo odvisnost od konkretne računalniške infrastrukture in omogočajo enostavno integracijo z obstoječimi sistemi in storitvami. Kljub temu pa nekatere platforme za razvoj aplikacij z malo ali nič programiranja še vedno zahtevajo določen nivo programerskih znanj, hkrati pa omogočajo manj možnosti kot orodja, ki prvenstveno temeljijo na programiranju. Zaradi tega orodja in platforme LCNC niso najprimernejše za razvoj kompleksnih rešitev, prinašajo pa tudi tveganje glede odvisnosti od ponudnika (Hakimi, 2019).

Obstaja pomembna razlika med pristopi in pripadajočimi orodji razvoja z malo in pristopi ter orodji razvoja z nič programske kode. Pri izbiri moramo biti pozorni in se zavedati pripadajočih prednosti in slabosti, saj bodo le na ta način orodja uresničila naša pričakovanja. V tabeli 1 so prikazane razlike med platformami, pri čemer je razvidno, da je vsak pristop namenjen določenemu krogu končnih uporabnikov (Hakimi, 2019).

Tabela 1: Razlike med platformami za razvoj z malo kode in platformami za razvoj z nič kode (Hakimi, 2019).

	Malo kode (low-code)	Nič kode (no-code)
Ciljni uporabniki	razvijalci in napredni uporabniki	poslovni uporabniki
Glavni cilj	hitrost razvoja	preprostost uporabe
Količina kode	malo, a obstaja	nič
Prilagajanje	večina komponent se lahko prilagodi	možnost le prilagajanje obstoječih predlog
Kompleksnost aplikacij	možnost ustvarjanja kompleksnejših aplikacij	lahko ustvari le preproste aplikacije
Stroškovna učinkovitost	učinkovito za podjetja z obstoječo ekipo razvijalcev	učinkovito za podjetja z manj razvito IT ekipo in velikimi pričakovanji
Vežanost na določeno platformo	prost prehod	včasih odvisno

1.2 Namen raziskave

Ker se omenjena orodja in pristopi vse bolj uveljavljajo v praksi, bomo v sklopu prispevka predstavili rezultate raziskave, katere namen je bil ugotoviti, kakšna je razširjenost in sprejetost tega pristopa v slovenskem prostoru ter katere pridobitve lahko pričakujemo ob uspešni vpeljavi in uporabi orodij za razvoj aplikacij z nič ali malo programiranja. V okviru raziskave smo se osredotočili na zaznane prednosti in pridobitve orodij za razvoj aplikacij z malo ali nič programiranja in preverili zaznavanje prednosti med strokovnjaki v slovenskih podjetjih in študenti. Želeli smo ugotoviti, v kolikšni meri poznajo ter uporabljajo orodja za razvoj aplikacij z malo ali nič programiranja. Izvedli smo tudi študijo primera ter tako preučili zaznano uporabnost in sprejetost enega izmed vodilnih orodij na tem področju.

1.3 Potek raziskave in uporabljene raziskovalne metode

Predstavljeno raziskavo smo izvedli v več sekvenčnih korakih. V prvem koraku smo za pridobitev vpogleda v rezultate dosedanjih raziskav na področju sprejetosti pristopa in orodij za razvoj aplikacij z malo ali nič programiranja izvedli podrobni pregled literature v bazah znanstvenih prispevkov IEEEExplore, ACM Digital Library, SpringerLink in ScienceDirect. Identificirali in analizirali smo tudi relevantne strokovne vire, ki povzemajo rezultate anket o razširjenosti pristopa LCNC. Na osnovi zbranih informacij smo nato oblikovali prvo verzijo vprašalnika, katerega namen je bil pridobitev vpogleda v stanje sprejetosti, poznavanja in pričakovanj glede pristopa LCNC v slovenskem okolju. Odgovori na večino vprašanj zaprtega tipa so bili oblikovani skladno s petstopenjsko Likertovo lestvico. Vprašalnik je obsegal tudi vprašanja povezava s profili anketirancev, kar nam je v koraku analize omogočalo podrobnejše raziskave, kot na primer analizirati vpliv predhodnih znanj o razvoju informacijskih rešitev na dožemanje pomena in pridobitev pristopa LCNC.

K sodelovanju v raziskavi smo povabili predstavnike slovenskih podjetij ter pridobili 78 v celoti izpolnjenih odgovorov iz različnih domen poslovanja. V študijskem letu 2019/2020 smo prvič v sklopu rednega študijskega procesa pri predavanjih in vajah sistematično predstavili in naslovili tematiko razvoja aplikacij z malo ali nič programiranja, pri čemer je 35 študentov višjih letnikov informatike pridobilo

konkretne izkušnje z gradnjo aplikacij z izbranim predstavnikom orodij LCNC. Po zaključenem razvoju so izpolnili oblikovano spletno anketo, v katero so bila, razen vprašanj o poznavanju, razširjenosti in pričakovanjih glede pristopa LCNC, vključena tudi standardizirana vprašanja modelov za oceno uporabnosti. Preliminarne rezultate izvedene raziskave smo povzeli in predstavili v prispevku (Beranič, Rek, & Heričko, 2020).

Preliminarno študijo in trende sprejetosti ter odnosa do uporabnosti orodij LCNC smo nadaljevali in nadgradili s podatki zbranimi v naslednjih dveh študijskih letih, in sicer v letu 2020/21 ter v letu 2021/22. Skupni vzorec predstavljene raziskave tako obsega 131 udeležencev. S pomočjo dodatnih analiz smo rezultate preliminarne analize nagradili z apliciranjem uveljavljenih modelov vrednotenja uporabnosti, kot sta SUS (angl. System Usability Scale) in UMUX (angl. Usability Metric for the User eXperience) ter modela TAM (angl. Technology Acceptance Model). S primerjalno analizo smo ugotavljali ne le skladnost rezultatov med strokovnjaki v podjetjih in študenti, temveč tudi primerljivost rezultatov z ugotovitvami relevantnih sorodnih raziskav v tujini. Nenazadnje pa smo raziskali, ali se rezultati skozi leta spreminjajo ter kako vplivajo izkušnje s konvencionalnim razvojem aplikacij, ki pretežno temelji na programiranju, na sprejetost, dožeto uporabnost in pričakovanja glede pristopa in orodij LCNC.

1.4 Struktura prispevka

V drugem poglavju prispevka so predstavljene relevantne sorodne raziskave na področju razvoja aplikacij z nič ali malo programiranja. Sledi poglavje z ugotovitvami in analizo mnenj študentov, ki temeljijo na praktičnih izkušnjah uporabe izbranega orodja pri razvoju konkretne rešitve. Predstavljene so tudi podatki o oceni uporabnosti in sprejetosti orodja PowerApps, za kar smo uporabili uveljavljene modele za oceno uporabnosti (SUS in UMUX) in sprejetosti (model TAM). Sledi četrto poglavje s podrobnejšim opisom rezultatov poznavanja in razširjenosti orodij za razvoj z malo ali nič programiranja v slovenskih podjetjih. V zaključku povzemamo ključne ugotovitve in le te primerjamo s svetovnimi trendi.

2 SORODNE RAZISKAVE

Pregled strokovne in znanstvene literature kaže na pomembno rast raziskovalnega zanimanja na podro-

čju razvoja aplikacij z malo ali nič programiranja. Zasedati je mogoče številna poročila, razprave in raziskave, ki naslavljajo izzive, povezane z vpeljavo in uporabo omenjenega pristopa razvoja. Na spletu se je v zadnjih petih letih zanimanje za orodja in pristope LCNC povečalo za 15x (Low-code development platform, 2022), v bazi ScienceDirect se je število objav na temo platform in pristopa LCNC v zadnjih petih letih močno povečalo, izstopajo pa tudi razprave iz področja razvoja aplikacij z malo ali nič programske kode (Al Alamin et al., 2021). Glede na napovedi naj bi bilo do leta 2024 več kot 65% aplikacij razvitih s pomočjo orodij za razvoj aplikacij z malo ali nič programiranja (Vincent et al., 2019; OutSystems, 2019). Gartnerjev kvadrant med najvidnejše predstavnike na tem področju postavlja rešitve podjetij Microsoft, Mendix, Salesforce, OutSystems in ServiceNow (Wong, Iijima, Leow, Jain, & Vincent, 2021). Orodje Microsoft Power Apps je strateško usmerjeno k poslovnim uporabnikom brez poglobljenih znanj in veščin programiranja, hkrati pa ponuja številne možnosti za razširljivost in vključevanje metod umetne inteligence (Wong et al., 2021).

Priljubljenost orodij za razvoj aplikacij z malo ali nič programiranja lahko neposredno povežemo tudi z naraščajočo pomembnostjo digitalne preobrazbe (Bloomberg, 2017; Hecht, 2021). Glede na raziskavo (Hecht, 2021) je prav digitalna preobrazba ob zviševanju odzivnosti v podjetju najpogostejši razlog za uporabo orodij za razvoj z malo ali nič programiranja. Platforme temeljijo na grafičnih uporabniških vmesnikih in vizualni abstrakciji, ki zahteva minimalno uporabo programske kode ter s tem tudi sodelujočim brez znanja programiranja omogoča razvoj funkcionalno delujočih aplikacij (Waszkowski, 2019; Sahay, Indamutsa, Di Ruscio, & Pierantonio, 2020). Orodja se uporabljajo v najrazličnejših industrijah, od področja komunalnih storitev in energije, do področja vlade in izobraževanja (OutSystems, 2019), in nena zadnje v proizvodni industriji. V omenjeni industriji so orodja kot omogočevalce digitalne transformacije preučili Sanchis et al (Sanchis, García-Perales, Fraile, & Poler, 2020). Na trgu je dostopna množica orodij, pri čemer se le-ta med seboj razlikujejo glede na ponujene funkcionalnosti. Sahay et al. so v raziskavi (Sahay et al., 2020) le-te identificirali ter organizirali z namenom ponujanja objektivnih značilnosti odločevalcem in potencialnim uporabnikom. Analizirali so osem predstavnikov orodij in jih med seboj primerja-

li glede na oblikovano taksonomijo. Sedem orodij sta med seboj primerjala tudi Bock and Frank (Bock & Frank, 2021) z uporabo šestih kategorij.

Priljubljenost pristopa in pripadajočih orodij za razvoj aplikacij z malo ali nič programiranja nakazujejo tudi številne raziskave trga. Glede na raziskavo podjetja OutSystems (OutSystems, 2019) kar 41 % sodelujočih podjetij že uporablja katero od platform za razvoj aplikacij z malo kode, medtem ko ima dodatnih 10 % namen tak pristop uporabiti v bližnji prihodnosti. Največ, kar 69 % primerov uporabe je s strani profesionalnih IT razvijalcev, nekoliko manj, 44 %, pa je poslovnih uporabnikov v sodelovanju z IT strokovnjaki. Orodja za razvoj brez programske kode so manj razširjena. Raziskava nakazuje, da se na tak način razvite aplikacije uporabljajo tako za namene interne uporabe med zaposlenimi, kot tudi za končne stranke podjetij. Stanje poznavanja pristopa je bilo naslovljeno tudi v raziskavi (Hedau & Mandge, 2021).

Razvoj aplikacij z malo ali nič programiranja prinaša številne prednosti. Uporaba lahko pohitri razvojni proces in posledično zmanjša stroške (Ploder, Bernsteiner, Schlögl, & Gschliesser, 2019). Ob tem tudi zmanjša odvisnost od IT razvijalcev in omogoča vključevanje drugih oddelkov v razvojni proces (Vincent et al., 2019; Ploder et al., 2019; Pantelimon, Rogojanu, Braileanu, Stanciu, & Dobre, 2019). Po mnenju (Jain, Iijima, Leow, Wong, & Vincent, 2021; Vincent et al., 2019; Wong et al., 2021) so primerni za širok spekter uporabe in sicer za gradnjo in vzdrževanje sodobnih poslovnih aplikacij, avtomatizacijo delovnih tokov in za sodelovalni razvoj. Platforme za razvoj aplikacij z malo kode se lahko po ugotovitvah (Wong et al., 2021) uporabijo tudi za razvoj poslovnih aplikacij, ki zahtevajo in potrebujejo visoko zmogljivost, razpoložljivost in razširljivost, varnost in API dostop za storitve podjetja in drugih oblačnih storitev.

Luo et al. (Luo, Liang, Wang, Shahin, & Zhan, 2021) so s pomočjo mnenj uporabnikov med drugim preverili tudi katere so glavne pridobitve uporabe LCNC orodij. Glede na odgovore, uporaba orodij za razvoj aplikacij z malo ali nič programske kode omogoča hitrejši razvoj in posledično hitrejšo izdajo na trg, izboljšana je sistemska kakovost in zmožnost integracije in razširitve. Kot izpostavljajo so orodja enostavna za uporabo z bogatimi in za uporabo pripravljenimi enotami. Nekatere raziskave so nasloville tudi faktorje, ki vplivajo na sprejetost pristopa in

orodij LCNC. Le-to so raziskovali na primer Alsaadi et al. (Alsaadi et al., 2021). S pomočjo vprašalnika izvedena raziskava je pokazala, da nivo znanja in izkušnje z razvojem programske opreme vplivajo na odločitev o uporabi omenjenih orodij. Ploder et al. (Ploder et al., 2019) pa so na drugi strani ugotovili, da zaznana uporabnost orodij za razvoj aplikacij z malo kode bistveno vpliva na namero o uporabi. Ugotovili so tudi, da so takšna orodja zelo prilagodljiva in uporabna za številne aplikacije in ne zgolj za en sistem, ki služi določenemu namenu. 35,8 % anketiranih v njihovi anketi je že uporabljalo LCNC orodja, ob tem pa so ugotovili, da zaznana uporabnost orodij pozitivno vpliva na namero za uporabo, medtem ko zaznana preprostost uporabe nima učinka (Ploder et al., 2019).

3 POZNAVANJE IN SPREJETOST ORODIJ ZA RAZVOJ APLIKACIJ Z MALO ALI NIČ PROGRAMIRANJA MED ŠTUDENTI

V okviru raziskave smo v vzorcu sodelujočih študentov informatike zbrali podatke o njihovem poznavanju, uporabi in mnenju glede prihodnosti pristopa razvoja aplikacij z malo ali nič programiranja. Dodatno so nas zanimale tudi njihove ocene uporabnosti in sprejetosti konkretnega orodja, ki so ga spoznali in uporabili v sklopu študijskega procesa. V raziskavo so bile vključene tri generacije študentov, pri čemer smo njihove odzive in mnenja, ki temeljijo tako na teoretičnih, kot pridobljenih praktičnih izkušnjah, pridobili s pomočjo sistematično oblikovanega vprašalnika. Poudariti velja, da gre za študente z dobrim poznavanjem vseh vidikov razvoja informacijskih rešitev, vključno s programiranjem, podatkovnimi bazami, spletnimi tehnologijami in programskim inženirstvom. Zbrani podatki vključujejo mnenja in ocene skupno 131 študentov prvega letnika magistrskega študijskega programa Informatika in tehnologije komuniciranja v študijskih letih 2019/2020, 2020/2021 in 2021/2022.

V prvem delu vprašalnika so sodelujoči s pomočjo pet-stopenjske Likertove lestvice (Joshi, Kale, Chandel, & Pal, 2015) izvedli samoocenitev poznavanja pogosto uporabljenih programskih jezikov ter poznavanja pristopa razvoja z malo ali nič programiranja. Glede na rezultate, večina študentov dobro pozna vsaj dva visokonivojska programska jezika. Večina študentov ima vsaj dve leti izkušenj s programiranjem, primarno zaradi svojega dodiplomskega

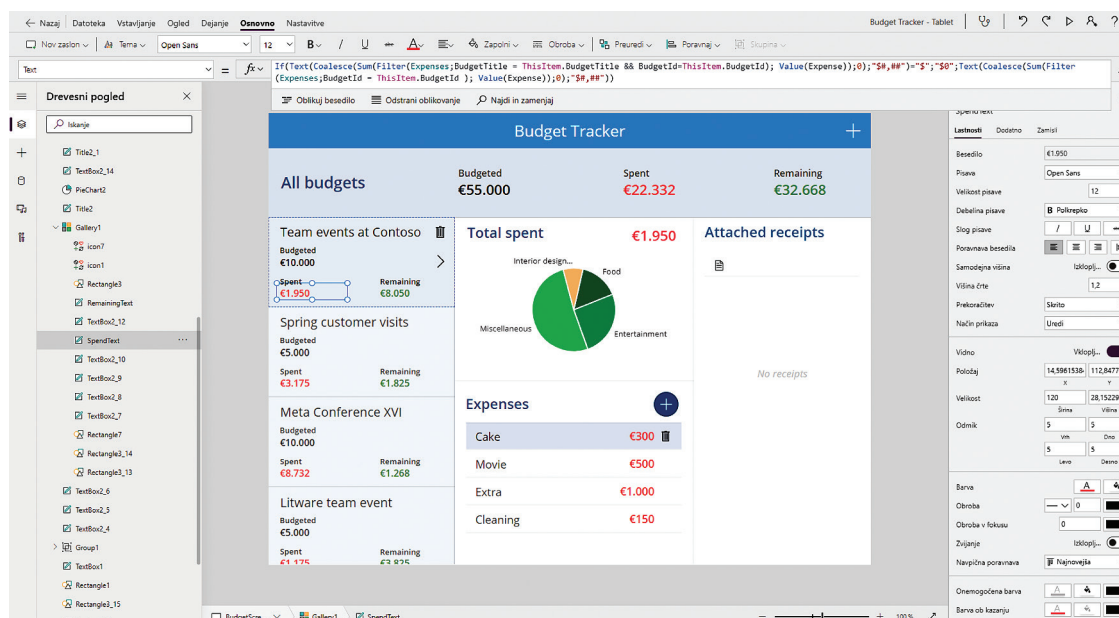
študija, medtem ko imajo študentje v povprečju 3,6 let izkušenj. 52,3 % študentov pristopa razvoja z malo ali nič programiranja pred obravnavo le-tega v študijskem procesu ni poznalo, medtem ko jih je 32,1 % že slišalo za tak pristop in orodja. Preostali vprašani so ta orodja že poznali. Rezultati sovpadajo z rezultati raziskave izvedene med podjetji, še posebej z rezultati poznavanja pridobljenimi med tehničnimi vlogami. Večina, kar 91 % študentov, je odgovorila, da orodij za razvoj aplikacij z malo ali nič programske kode v poslovnih okoljih do sedaj še niso uporabljali. Odstotek se razlikuje glede na rezultate v slovenskih podjetjih. Pri tehničnih vlogah v podjetjih ta vrednost znaša 80 %, medtem ko je med vodstvenimi vlogami 65 % takšnih, ki orodij za razvoj z malo ali nič programske kode še niso uporabljali. Poznavanje orodij za razvoj z malo ali nič kode je v vseh treh sodelujočih generacijah študentov podobno. Delež študentov, ki so že uporabljali takšna orodja se namreč z leti ni pomembno spremenil.

3.1 Izbrano orodje za razvoj aplikacij z malo ali nič programiranja

Študenti, ki so sodelovali v študiji, so uporabljali Microsoftovo platformo za razvoj z malo kode Power Apps. Power Apps je komplet orodij, ki je na voljo naročnikom storitve Office 365. Gre za zbirko aplikacij, storitev in konektorjev ter podatkovno platformo, ki sestavlja okolje za hiter razvoj aplikacij. S pomočjo Power Apps je mogoče zgraditi poslovne aplikacije, ki se povežejo na podatke, shranjene v podatkovni platformi Microsoft Dataverse ali v drugih podatkovnih virih, kot so SharePoint, Microsoft 365, Dynamics 365 ali SQL Server.

Izbrano orodje je v Gartnerjevem kvadrantu pozicionirano med vodilne platforme za razvoj aplikacij z malo kode. Orodje skupaj s Power Automate in Dataverse tvori celovito platformo Power Platform, platformo za razvoj aplikacij z malo kode. Glede na rezultate raziskave se Power Apps trenutno uporablja predvsem za razvoj preprostih spletnih in mobilnih vmesnikov za interne aplikacije. Razvoj orodja je potekal od namembnosti individualnim, neizkušenim, razvijalcem, do trenutne usmeritve k večjim ekipam in IT uporabnikom (Jain et al., 2021; Wong et al., 2021).

Z uporabo orodja je omogočen razvoj odzivnih aplikacij, ki lahko tečejo v brskalniku ali na mobilnih ter tabličnih napravah. Primer izdelave aplikacije v



Slika 1: Razvoj aplikacije s pomočjo orodja Microsoft Power Apps.

orodju Microsoft Power Apps je prikazan na sliki 1. V okviru orodja je mogoč razvoj aplikacij, katerih funkcije nastajajo brez ali z minimalno uporabo kodiranja. Power Apps omogočajo povleci in spusti vmesnik za gradnjo aplikacij, naprednejši razvijalci pa lahko aplikacije razširijo z .NET razvojem kompleksnejše logike, integracijo podatkov in uporabniškim vmesnikom po meri. Microsoft zavoljo licence OpenAI za uporabo sistema GPT-3 za umetno inteligenco in uporabo v jeziku Power FX, omogoča preprosto generiranje povpraševanj in logike skozi stavke v naravnem jeziku. Orodje Power Apps ponuja tudi številne varnostne funkcije, kot je integracija z Active Directory, pravila za preprečevanje izgube podatkov in šifriranje podatkov. Za sočasen razvoj se lahko razvijalci poslužijo komponentno zasnovanega razvoja in razvojnih orodij, kot sta Azure DevOps in GitHub (Jain et al., 2021; Wong et al., 2021)

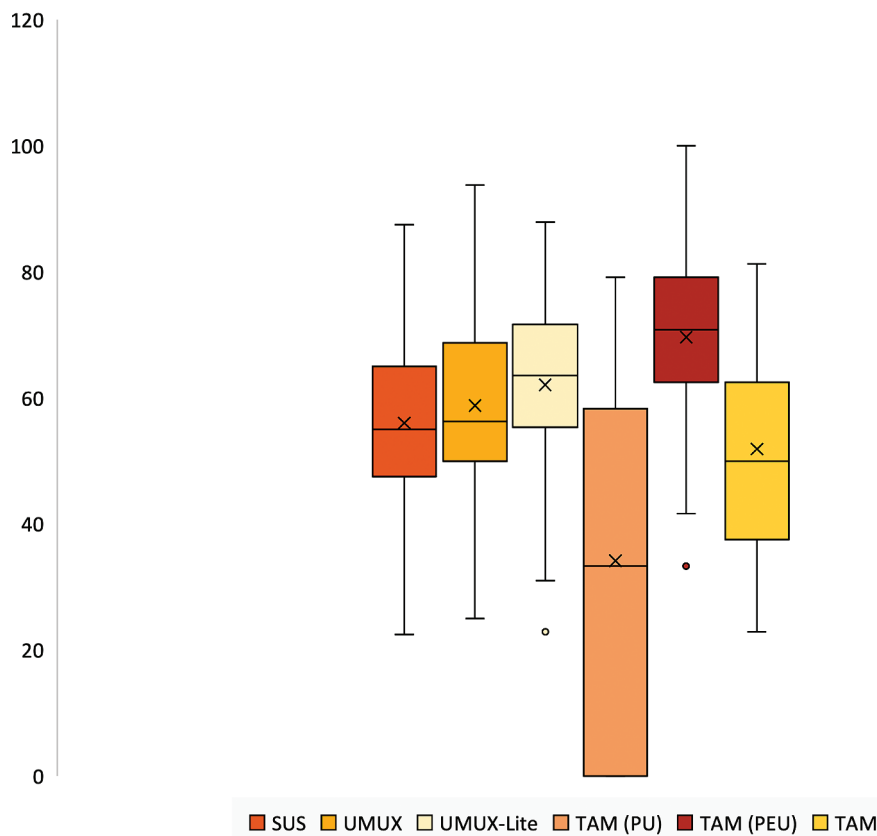
3.2 Zaznana uporabnost orodja za razvoj aplikacij z malo ali nič programiranja

V okviru raziskave med študenti smo izvedli tudi študijo uporabnosti orodja Power Apps, ki so ga, kot tipičnega predstavnika platforme za razvoj aplikacij z malo ali nič programiranja, študentje spoznali in uporabili v okviru študijskega procesa. Uporabnost je bila ovrednotena z uporabo uveljavljenega pristopa merjenja uporabnosti. S pomočjo desetih vprašanj, ki sestavljajo metodo SUS (angl. system Usability Scale), ter ocene le-teh s pomočjo petstopenjske lestvice, lahko pridobimo vpogled v subjektivna zaznavanja uporabnosti ocenjevanega sistema (Brooke, 2013). Na podlagi odgovorov smo izračunali vrednost SUS za vsakega študenta skladno z uveljavljenim koraki (Brooke, 2013). Najvišji možni SUS rezultat je 100, povprečni SUS rezultat pri anketirani populaciji študentov pa je znašal 54,04 s standardnim odklonom 12,68 in mediano 55. Rezultate prikazuje tabela 2.

Tabela 2: Rezultati metrik SUS, UMUX, UMUX-Lite, PU, PEU in TAM za uporabnost in sprejetost platform glede na odgovore anketiranih študentov.

	Povprečje	Medijana	Modus	Standardni odklon
SUS	56,04	55	50	12,68
UMUX	58,79	56,25	50	14,95
UMUX-Lite	62,09	63,53	71,65	13,58
PU*	34,17	33,33	0	26,29
PEU*	69,64	70,83	75	13,96
TAM*	51,90	50	62,50	15,57

* Pri podatkih za PU, PEU in TAM je bil uporabljen manjši vzorec (35 študentov).



Slika 2: Prikaz rezultatov metrik za uporabnost in sprejetost orodja Microsoft Power Apps.

Obstajajo različne interpretacije rezultata SUS vprašalnika. Glede na (Brooke, 2013) se lahko rezultat 54,04 tolmači kot sprejemljiv oz. predstavlja oceno F, ki je na spodnji meji sprejemljivosti (Brooke, 2013). Glede na odzive študentov je za tiste z nižjim predznanjem programiranja uporabnost večja, kar potrjuje ostale raziskave.

Dodatno smo na osnovi mnenj sodelujočih študentov za oceno uporabnosti uporabili še t.i. Metriko uporabnosti za uporabniško izkušnjo UMUX (angl. usability metric for the user experience) in njeno preprostejšo naslednico UMUX-Lite. UMUX vprašalnik sestavljajo štiri vprašanja, s pomočjo katerih UMUX meri komponente učinkovitosti, zmogljivosti in zadovoljstva (Finstad, 2010). Podobno kot pri metodi SUS, kot rezultat pridobimo vrednost med 0 in 100. UMUX-Lite predstavlja poenostavljen UMUX s samo dvema vprašanjem (Lewis, Utesch, & Maher, 2013). Tako smo v vprašalnik naše raziskave razen desetih trditvev pristopa SUS vključili še vse štiri trditve, ki predstavljajo osnovo za izračun vrednosti UMUX in UMUX-Lite. Kot je razvidno iz tabele 2 je povprečni izračunani rezultat pri UMUX podoben kot pri SUS,

to je 58,79 (Finstad, 2010; Lewis et al., 2013). Povprečna vrednost po metriki UMUX-Lite je 62,09. Rezultati izvedenih študij uporabnosti so povzeti v tabeli 2 ter prikazani na sliki 2.

Za dodatno analizo odnosa do sprejetosti izbranega orodja LCNC smo uporabili še model TAM (angl. technology acceptance model). Glede na model je zaznana uporabnost in zaznana preprostost uporabe izhajajoča iz lastnosti sistema. Uporabnost in preprostost uporabe pri uporabnikih sprožita odnos do uporabe sistema in kasneje dejansko uporabo. Zaznana uporabnost (PU – angl. perceived usefulness) je mera, do katere oseba verjame, da bi uporaba določenega sistema izboljšala učinkovitost pri delu, zaznana preprostost uporabe (PEU – angl. perceived ease of use) pa je stopnja, do katere oseba verjame, da bi bila uporaba določenega sistema brez napora (Davis, 1993). Anketiranim študentom smo za ocenjevanje zaznane uporabnosti (PU) postavili trditve po Davisovem modelu (Davis, 1993).

Da smo rezultate številčno ovrednotili in izračunali zaznano uporabnost, smo odgovore iz Likertove lestvice ovrednotili od 0 do 4 in izračunali odstotno

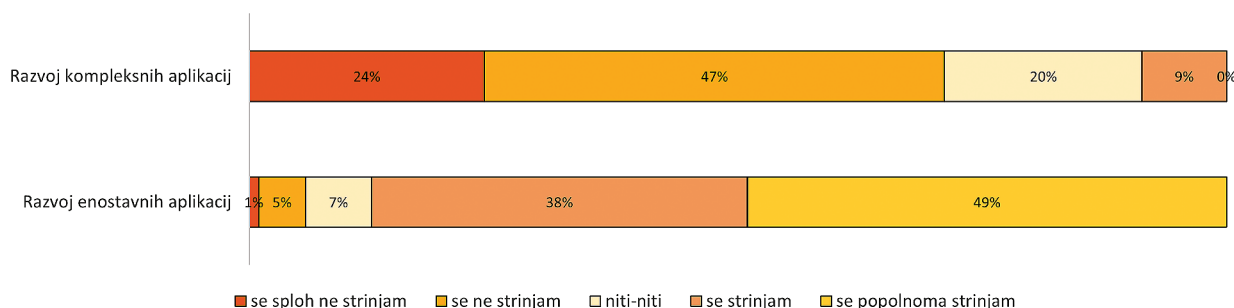
vrednost. Izračunana povprečna vrednost PU pri anketiranih študentih znaša 34,17 % z mediano pri 33,33 % in standardnim odklonom 26,29. Pri metriki PU smo zaznali negativno korelacijo med zaznano uporabnostjo in povprečnim poznavanjem programskih jezikov. Ta podatek podpira trditve, da je zaznana uporabnost orodij za razvoj aplikacij z malo ali nič programiranja višja pri uporabnikih z manj izkušnja mi s programiranjem.

Za ocenjevanje zaznane preprostosti uporabe (PEU) smo podoben izračun izvedli na podlagi temu

namenjenih trditve (Davis, 1993). Rezultati so bili tudi tukaj podobni. Povprečna vrednost PEU je znašala 69,64 % z mediano 70,83 %. Iz pridobljenih podatkov za PU in PEU smo iz povprečja izračunali ocenjeno vrednost odnosa do uporabe orodja. Ta v našem primeru znaša v povprečju 51,9 % z mediano 50,0 % in negativno korelacijo s poznavanjem programskih jezikov. S temi podatki lahko ugotovimo, da je polovična sprejetost in pripravljenost za uporabo, kar sovпада tudi z rezultati ankete pri podjetjih. Tudi rezultati vseh TAM metrik so prikazani v tabeli 2.

Tabela 3: Zaznane prednosti uporabe orodij za razvoj z malo ali nič programiranja

	1	2	3	4	5	Povprečje	Št.dev.
Hitrejši razvoj	3 %	14 %	17 %	49 %	16 %	3,62	1,01
Razvoj tudi s strani poslovnih uporabnikov	6 %	15 %	23 %	47 %	9 %	3,38	1,03
Enostavne prilagoditve in spreminjanje	1 %	10 %	34 %	48 %	8 %	3,52	1,01
Stroškovno učinkovitejši razvoj	5 %	14 %	27 %	43 %	11 %	3,40	0,81
Razvoj brez poglobljenega znanja programiranja	1 %	13 %	24 %	39 %	23 %	3,71	0,99
Boljšo odzivnost na spremembe	4 %	26 %	41 %	26 %	3 %	2,97	0,90
Manj napak v implementaciji	3 %	22 %	30 %	35 %	10 %	3,28	0,95
Boljše sodelovanje z naročnikom	3 %	19 %	32 %	39 %	7 %	3,29	1,01



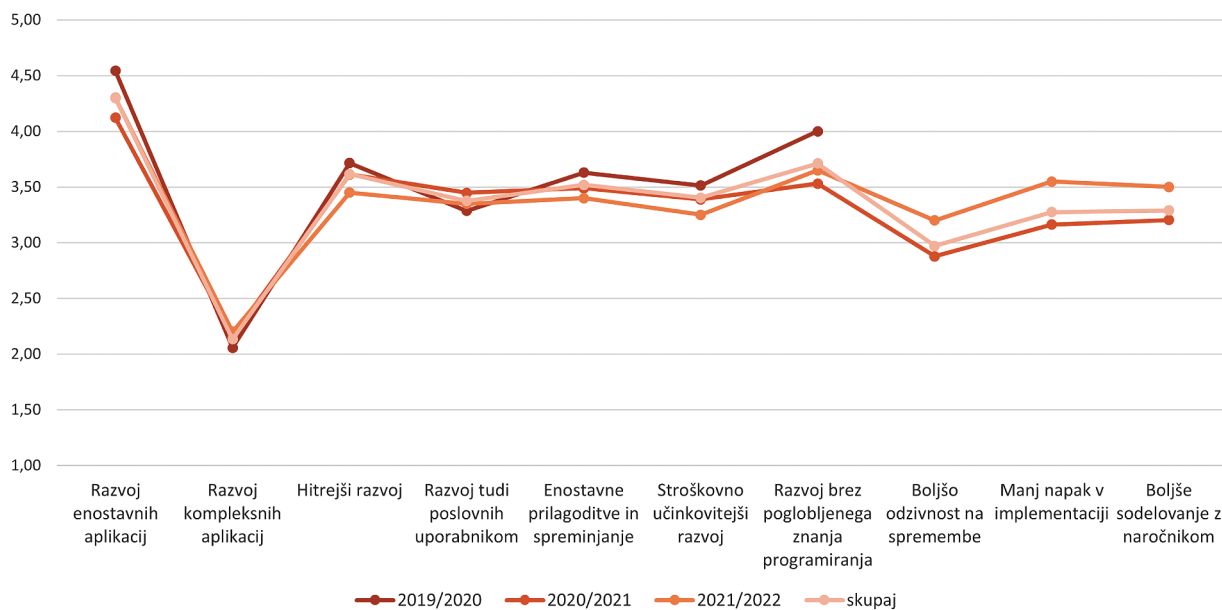
Slika 3: Zaznana primernost uporabe orodij za razvoj z malo ali nič programiranja

3.3 Prednosti in pričakovanja glede pristopa razvoja aplikacij z malo ali nič programiranja

Sovpadajoč z zaznanimi prednostmi in pridobitvami uporabe orodij za razvoj z malo ali nič programiranja, zaznanimi v sorodnih študijah, smo tudi študente povprašali glede njihovega zaznavanja le-teh. Študenti so ovrednotili podane trditve s pomočjo pet-stopenjske Likertove lestvice, pri čemer 1 pomeni se sploh ne strinjam in 5 pomeni se popolnoma strinjam. Rezultati ovrednotenih prednosti uporabe orodij za razvoj aplikacij z malo ali nič programiranja

so prikazani v tabeli 3. Na sliki 3 so prikazane zaznane primernosti uporabe orodij za razvoj z malo ali nič programiranja za primere razvoja enostavnih in kompleksnih aplikacij. V primeru kompleksnih aplikacij se nihče od anketirancev s primernostjo popolnoma ne strinja.

Večina študentov, 49 %, se popolnoma strinja s trditvijo, da so orodja za razvoj aplikacij z malo ali nič programiranja primerna za razvoj enostavnih aplikacij. Temu je z izbiro odgovora se strinjam pritrldilo še dodatnih 38 % vprašanih. Posledično se večina



Slika 4: Primerjava zaznanih prednosti med generacijami študentov.

vprišanih, 71 %, ne strinja s tem, da so orodja LCNC primerna za razvoj kompleksnejših aplikacij. Kot je razvidno iz tabele 3, kot prednost uporabe orodij za razvoj z malo ali nič programiranja vidijo v možnosti razvoja tudi poslovnih uporabnikov in možnostjo razvoja brez poglobljenega znanja programiranja. Tudi sicer so študenti kot glavne prednosti uporabe orodja Power Apps prepoznali intuitiven in preprost uporabniški vmesnik, samodejno ustvarjanje zaslonov na osnovi tabel ter preprosto integracijo s podatkovno bazo, kar vsekakor omogoča hitro implementacijo preprostih aplikacij. Kot eno izmed prednosti vidijo tudi boljše sodelovanje z naročnikom, kar je potrdilo 46% vprišanih. Med zaznanimi pomanjkljivostmi orodja so študenti izpostavili predvsem slabše odzivne čase orodja, pogrešajo pa tudi podrobnejšo dokumentacijo ter večjo skupnost razvijalcev z aktivnimi spletnimi forumi, kjer bi lahko poiskali odgovore oz. rešitve za svoje izzive.

V okviru raziskave smo primerjali tudi odgovore študentov različnih generacij. Rezultate zaznanih prednosti uporabe orodij za razvoj z malo ali nič programiranja ter zaznane primernosti uporabe orodij za razvoj z malo ali nič programiranja v generacijah 2019_2020, 2020_2021 in 2021_2022, prikazuje slika 4. Kot je razvidno, se povprečne vrednosti ocenjenih trditev med generacijami bistveno ne razlikujejo.

Podobno kot to so to analizirale sorodne študije ter del naše raziskave namenjene podjetjem, smo

tudi pri sodelujočih študentih zbrali in analizirali njihova pričakovanja glede prihodnosti razvoja poslovnih aplikacij z malo ali nič programiranja. S povprečnim rezultatom 2,8 so pričakovanja študentov nekoliko pesimistična in ne sovpadajo s pričakovanji strokovnjakov v poslovnih okoljih. V slovenskih podjetjih je sicer trenutna uporaba obravnavanega pristopa razvoja aplikacij precej nizka, kljub temu pa velik del sodelujočih verjame v pozitivne napovedi glede LCNC orodij in pristopov.

4 POZNAVANJE IN RAZŠIRJENOST PRISTOPA IN ORODIJ ZA RAZVOJ APLIKACIJ Z MALO ALI NIČ PROGRAMIRANJA V SLOVENSkih PODJETJIH

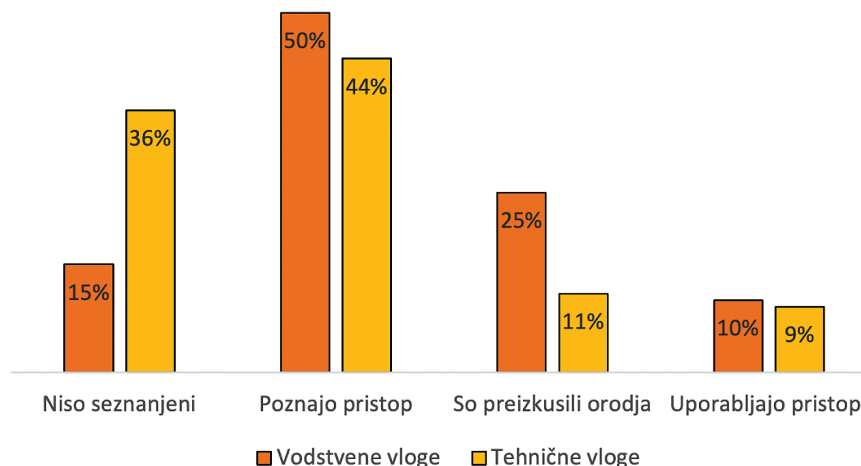
Raziskava je obsegala tudi anketo izvedeno med slovenskimi podjetji. Vprašalnik smo oblikovali glede na cilje raziskave izhajajoč iz pregleda literature in sorodnih strokovnih prispevkov. Tudi v primeri raziskave med podjetji smo v prvem delu vprašalnika najprej preverili profil sodelujočih, natančneje umešitev samega podjetja glede na velikost, število zaposlenih na področju informatike, število razvijalcev in področje delovanja. V nadaljevanju vprašalnika pa smo od podjetij želeli pridobiti informacije o predhodnem poznavanju tehnologij za razvoj aplikacij z malo ali nič programiranja, obstoječi integriranosti takšnih tehnologij v podjetje, scenarijih uporabe, razlogih za uporabo ali neuporabo in oceni same uporabe orodij.

Glede na pozitivne trende, povezane z uporabo obravnavanih orodij, smo želeli preveriti, v kolikšni meri to velja tudi za slovenska podjetja. Za pridobitev podatkov smo vabilo in povezavo do spletnega vprašalnika poslali različnim predstavnikom podjetij, pri čemer smo prejeli 78 popolnih odgovorov. Med sodelujočimi podjetji jih je večina, 34,6 %, predstavnikov manjših podjetij z med 11 in 50 zaposlenimi, 28,2 % sodelujočih je bilo iz podjetij z več kot 250 zaposlenimi, 25,6 % pa srednje velikih podjetij med 51 in 250 zaposlenimi. Zgolj manjši del, 11,5 %, je prihajalo iz podjetij z manj kot 10 zaposlenimi. Za skoraj 70 % sodelujočih predstavlja razvoj programske opreme glavno področje poslovanja, kjer s 50 % sledi tehnologija in internet ter s 34,6 % svetovanje. 10,3 % sodelujočih podjetij deluje na področju energije, 9 % jih prihaja iz bančnega in finančnega sektorja, 6,4 % iz zdravstva in farmacije in 5,1 % iz zavarovalniškega sektorja. 26,9 % v raziskavo vključenih podjetij na področju informatike zaposluje med 50 in 100 ljudi, med katerimi jih ima 47,6 % zaposlenih med 20 in 50 razvijalcev. 16,7 % podjetij zaposluje več kot 100 ljudi na področju informatike, 19,3 % pa manj kot 10. 14,1 % udeležencev je vodij informatike ali glavnih tehnoloških direktorjev, 5,1 % pa jih ne dela na področju informacijske tehnologije. 10,3 % udeležencev je vodij na oddelku informatike v podjetju, medtem ko jih 25,6 % ima vlogo arhitektov programske opreme. 7,7 % jih je poslovnih analitikov, 6,4 % inženirjev kakovosti, 53,8 % pa jih je razvijalcev.

Poznavanje in uporabo pristopa razvoja z malo ali nič programiranja smo obravnavali ločeno med

tehničnimi in vodstvenimi vlogami. Rezultati so prikazani na sliki 5. 50 % sodelujočega vodstvenega kadra je odgovorilo, da so že zasledili in poznajo razvoj aplikacij z malo ali nič programiranja. Podobno je tudi pri tehničnih vlogah, kjer je enako odgovorilo 43,6 % vprašanih. 25 % sodelujočih iz vodstvenih vlog je v podjetju že preizkusilo orodja za razvoj z malo ali nič programiranja, a jih le 10 % že uporablja tak razvojni pristop. Med tehničnimi vlogami jih je zgolj 10,9 % že preizkusilo takšna orodja, podobno je tudi pri vodstvenih vlogah, z 9,1 %. Pri tehničnih vlogah glede na vodstvene vloge izstopa predvsem to, da kar 36,4 % vprašanih pred našo anketo za razvoj aplikacij z malo ali nič programiranja še ni slišalo.

20 % vprašanih vodilnih je odgovorilo, da v svoji IT strategiji že uporabljajo ali v bližnji prihodnosti načrtujejo uporabo orodij za razvoj z malo ali nič programiranja, medtem ko jih 60 % ne načrtuje uporabe. Med tistimi, ki že uporabljajo ta orodja, so ta v uporabi s strani vsaj petih zaposlenih. Ta orodja uporabljajo poslovni uporabniki, v enem primeru pa tudi profesionalni IT razvijalci. V podjetjih, ki načrtuje uporabo, bodo orodja prav tako uporabljali zgolj IT razvijalci. Ker večina sodelujočih ne načrtuje uporabe orodij za razvoj z malo ali nič programiranja, smo raziskali tudi razloge za to. Po mnenju kar 68,3 % vprašanih je eden ključnih razlogov pomanjkanje znanja in izkušenj o takšnem razvoju in pripadajočih platformah ter orodjih. Med ostalimi razlogi je skrb pred preveliko navezanostjo na določenega ponudnika ter dvomi o izvedljivosti ter primernosti omenjenega



Slika 5: Poznavanje in uporaba pristopa in orodij za razvoj aplikacij z malo ali nič programiranja (vodstvene vs. tehnične vloge).

pristopa pri razvoju njihovih aplikacij. Ta razlog je navedlo kar 75 % vprašanih. 43,8 % vprašanih je skrbela razširljivost takšnih aplikacij, 37,5 % pa varnost.

Med sodelujočimi, ki že uporabljajo pristop razvoja z malo ali nič programiranja ali to načrtujejo v bližnji prihodnosti, smo preverili njihove izkušnje in izpolnitev pričakovanj. 16,7 % vseh vprašanih sodi v to skupino. Rezultati ankete so prikazani v tabeli 4. Sodelujoči menijo, da je pristop LCNC primeren za razvoj spletnih in mobilnih aplikacij, ki se bodo uporabljale znotraj podjetja (povprečna ocena 4,0). Nekoliko nižjo oceno, 3,5, dosega ocena primernosti za razvoj aplikacij za zunanje stranke. Glede na odgovore je pristop za razvoj z malo ali nič programiranja primeren za avtomatizacijo ponavljajočih razvojnih nalog, kot na primer razvoj vnosnih obrazcev, nekoliko manj primeren pa za razvoj informacijske podpore poslovno kritičnih sistemov. Anketiranci ocenjujejo, da lahko uporaba razvojnega pristopa LCNC prihrani veliko časa, s čimer se strinja ali popolnoma strinja skoraj 85 % sodelujočih, ter da pomembno prispeva k uspešni digitalni preobrazbi podjetja, kar podpira 69,2 % sodelujočih.

Kot že omenjeno, obstajajo številna orodja in platforme za razvoj z malo ali nič programiranja. Med sodelujočimi podjetji, ki so že uvedla tak pristop razvoja ali to še načrtujejo v bližnji prihodnosti, je le manjši delež. Sodelujoče, ki nastopajo v vodstvenih vlogah, smo povprašali tudi o stanju t.i. informatike v senci v njihovem podjetju. Informatika oz. IT v senci (angl. Shadow IT) predstavlja programsko in strojno opremo, ki se uporablja in ni odobrena oz. pod nadzorom IT oddelka posameznega podjetja. Takšna oprema lahko predstavlja precejšnjo var-

nostno tveganje, ogroža zaupnost podatkov, predstavlja tveganje glede skladnosti za regulirane organizacije, hkrati pa negativno vpliva na možnost rednega vzdrževanja in posodabljanja programske opreme. Po drugi strani lahko nepoklicni razvijalci uporabljajo orodja, ki jih je sicer odobril IT oddelek, a če niso ustrezno upravljani, lahko tudi to predstavlja podobna tveganja kot IT v senci (OutSystems, 2019; Vincent et al., 2019). 50 % sodelujočih je mnenja, da IT v senci v njihovi organizaciji ne predstavlja tveganja, čeprav se zavedajo, da obstaja. 10 % vprašanih zaznava IT v senci v podjetju kot precejšnje tveganje, medtem ko 40 % meni, da v njihovem podjetju IT v senci ne obstaja. Podobno vprašanje je naslovilo tudi rešitve, ki so jih v podjetjih sprejeli glede IT-ja v senci in samostojnega razvoja s strani poslovnih uporabnikov. Zgolj 10 % jih je odgovorilo, da so izbrali orodja in platforme za razvoj z malo ali nič programiranja, ki so morala biti uporabljena s strani poslovnih uporabnikov za zmanjšanje tveganja. Preostali niso imeli aplikacij, ki bi jih razvijali poslovni uporabniki ali pa njihov IT oddelek ne upravlja IT v senci in usposabljanja, ali poskušajo upravljati te izzive z omejenimi sredstvi.

Glede na visoka pričakovanja, povezana z razvojem z malo ali nič programiranja, zaznana v številnih poročilih, smo preverili tudi mnenje v slovenskih organizacijah. Med vodstvenimi člani je bilo mnenje popolnoma deljeno: polovica jih meni, da bodo pozitivna pričakovanja izpolnjena in pričakovane pridobitve uresničene, polovica pa se s tem ne strinja oz. je skeptična. Pri bolj tehnično usmerjenih kadrih je zaznati nekoliko več dvomov - zgolj 41,8 % jih predvideva uresničitev pričakovanj. Posledično je prič-

Tabela 4: Mnenje udeležencev, ki že uporabljajo orodja za razvoj programske opreme z malo ali nič programiranja, ali načrtujejo uporabo v bližnji prihodnosti.

	1	2	3	4	5	Povprečje	Št.dev.
S pristopom LCNC je mogoče razvijati spletne in mobilne aplikacije za uporabo znotraj podjetja.	0 %	7,7 %	7,7 %	61,5 %	23,1 %	4,0	0,82
S pristopom LCNC je mogoče razvijati spletne in mobilne aplikacije za stranke izven podjetja.	7,7 %	15,4 %	7,7 %	61,5 %	7,7 %	3,5	1,13
Implementacija pristopa LCNC poveča hitrost razvoja aplikacij.	0 %	15,4 %	0 %	61,5 %	23,1 %	3,9	0,95
Uporaba pristopa LCNC lahko prihrani veliko časa.	0 %	0 %	7,7 %	76,9 %	15,4 %	4,1	0,49
Pristop LCNC je uporaben za avtomatizacijo ponovljivih razvojnih opravil, kot je razvoj vnosnih polj.	0 %	7,7 %	0 %	76,9 %	15,4 %	4,0	0,71
Pristop LCNC lahko v veliki meri prispeva k digitalni transformaciji podjetja.	0 %	0 %	15,4 %	53,8 %	15,4 %	3,7 %	0,95

kovan tudi rezultat glede ocene primernosti vpeljave pristopa LCNC - pri obeh vlogah prevladuje skepsa (45 % pri vodstvenih vlogah in 41,9 % pri tehničnih), temu sledi neodločenost (35 % pri vodstvenih in 30,9 % pri tehničnih vlogah), pozitiven odnos pa izkazuje 30 % v anketi sodelujočih vodilnih in 29,1 % tehničnih strokovnjakov. Zgolj 10 % vodstvenih delavcev in in 18,2 % tehničnih strokovnjakov je izrazilo negativen oz. odklonilen odnos do pristopa k razvoju aplikacij z malo ali nič programiranja.

5 SKLEP

Pospešen napredek sodobnih tehnologij odpira nove možnosti glede pristopov in orodij za hiter razvoj novih informacijskih rešitev. Mednje sodijo tudi pristop razvoja aplikacij z malo ali nič programiranja ter pripadajoča orodja. Le-ta naslavlja znane izzive, povezane s produktivnostjo in agilnostjo razvoja tako v oddelkih za informatiko kot tudi s poslovno vodenim razvojem aplikacij ter informatiko v senci, dodatno pa se nakazujejo kot idealna za hiter razvoj inovativnih rešitev v procesih digitalne preobrazbe ter za razvoj aplikacij brez poglobljenih znanj programiranja t. i. »citizen development«. V članku smo predstavili rezultate študije o poznavanju, sprejetosti in uporabnosti orodij za razvoj z malo ali nič programiranja v slovenskem okolju. Raziskava je vključevala 78 predstavnikov slovenskih podjetij ter 131 študentov magistrskega študijskega programa s področja informatike iz treh študijskih generacij.

Pridobljeni rezultati omogočajo vpogled v trenutno stanje na področju razvoja in orodij za razvoj z malo ali nič programiranja tako v poslovnem kot akademskem okolju. Ugotovili smo, da le majhen delež, približno 10%, organizacij v Sloveniji že uporablja omenjen razvojni pristop. Prav tako jih kar 60 % nima namena vključitve le-tega v svoje okolje v bližnji prihodnosti. Rezultati globalne raziskave (OutSystems, 2019) se od slovenskih precej razlikujejo, saj nakazujejo, da kar 41 % sodelujočih že uporablja pristop za razvoj z malo ali nič programiranja, velika večina preostalih pa ga namerava vpeljati. Kljub slabi uporabi pa so predstavniki slovenskih podjetij seznanjeni z orodji za razvoj z malo ali brez programiranja. Polovica vodstvenih in 44% tehničnih vlog pozna pristop, kar 25% delež vodstvenih vlog pa je orodja v svojem podjetju tudi že preizkusilo v praksi. Večji delež tistih, ki s pristopom niso seznanjeni, je v okviru tehničnih vlog, prav tako pa uporaba in

poznavanje obravnavanega pristopa razvoja ni veliko niti med študenti. Med glavne razloge neuporabe pristopa med slovenskimi podjetji lahko identificiramo dvom v izvedljivost gradnje aplikacij, ki jih potrebujejo in pomanjkanje znanja o razvoju z malo ali nič programiranja in povezanih orodjih ter platformah. Prav tako obstaja bojazen glede prevelike odvisnosti in vezanosti na zgolj enega ponudnika orodij LCNC. Najpogostejši trije pomisleki za vpeljavo torej sovpadajo z rezultati globalne raziskave (OutSystems, 2019).

Slabše poznavanje, razširjenost in omejeno uporabo tega pristopa v slovenskem okolju je zaznati tudi med študenti, saj je le 18 % študentov že uporabilo pristop ali orodja za razvoj aplikacij z malo ali nič programiranja. Orodja in pristop LCNC se običajno omejuje na razvoj preprostih poslovnih aplikacij za interne uporabnike. Zanimive so tudi ugotovitve, ki temeljijo na uporabi modelov SUS, UMUX, UMUX-Lite in TAM pri treh generacijah študentov, ki so pridobili praktične izkušnje s konkretnim predstavnikom orodja za razvoj z malo ali nič programiranja. Rezultat kaže na primerno zrelost in uporabnost orodij za razvoj z malo ali nič programiranja, hkrati pa je dojeta uporabnost in namera glede uporabe negativno korelirana s programerskimi znanji, kar delno potrjuje predvidevanja in pričakovanja, da so orodja LCNC še posebej uporabna in koristna za razvijalce brez poglobljenih programerskih znanj oz. manj izkušene razvijalce. Kljub vsem pridobitvam in prednostim pa lahko le-te dosežemo le v primeru, če k razvoju pristopamo sistematično. V ta namen že obstajajo procesna ogrodja, ki opredeljujejo vloge, aktivnosti in združujejo dobre prakse (White, 2004). Na osnovi zbranih empiričnih rezultatov in njihove primerjave s podobnimi študijami v tujini, lahko sklepamo, da uporaba orodij in pristopa razvoja aplikacij z malo ali nič programiranja v Sloveniji opazno odstopa od globalnih usmeritev. Ker pa anketiranci, tako v poslovnem kot univerzitetnem okolju, verjamejo v pozitivna pričakovanja, povezana s sodobnimi globalnimi trendi in z razvojem z malo ali nič kode, je tudi v Sloveniji v prihodnjih letih pričakovati porast zanimanja in večjo uporabo tega obetavnega pristopa.

LITERATURA

- [1] Al Alamin, M. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. B., & Iqbal, A. (2021). An empirical study of developer discussions on low-code software development challenges. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (msr)* (p. 46-57). doi: 10.1109/MSR52588.2021.00018
- [2] Alsaadi, H., Radain, D., Alzahrani, M., Alshammari, W., Alahmadi, D., & Fakieh, B. (2021, 09). Factors that affect the utilization of low-code development platforms: survey study. *Revista Română de Informatică, i Automatică*, 31, 123-140. doi: 10.33436/v31i3y202110
- [3] Beranič, T., Rek, P., & Heričko, M. (2020). Adoption and usability of low-code/no-code development tools. In V. Strahonja, W. Steingartner, & V. Kirinić (Eds.), *Ceciis [elektronski vir]: Central european conference on information and intelligent systems: proceedings: 31th international scientific conference, october 7- 9, 2020, Varaždin, Croatia* (p. 97–103). Faculty of Organization and Informatics. Retrieved from <http://archive.ceciis.foi.hr/app/public/conferences/2020/Proceedings/ETICT/ETICT2.pdf> (Nasl. z nasl. zaslona Opis vira z dne 4. 2. 2021 Bibliografija: str. 103 Abstract)
- [4] Bloomberg, J. (2017, Jul). *The low-code/no-code movement: More disruptive than you realize*. Forbes Magazine. Retrieved from <https://www.forbes.com/sites/jasonbloomberg/2017/07/20/the-low-codeno-code-movement-more-disruptive-than-you-realize/?sh=74ab7567722a>
- [5] Bock, A. C., & Frank, U. (2021). In search of the essence of low-code: An exploratory study of seven development platforms. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (models-c)* (p. 57-66). doi: 10.1109/MODELS-C53483.2021.00016
- [6] Brooke, J. (2013, 01). Sus: a retrospective. *Journal of Usability Studies*, 8, 29-40.
- [7] Davis, F. D. (1993). User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. *International Journal of Man-Machine Studies*, 38(3), 475-487. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0020737383710229> doi: <https://doi.org/10.1006/imms.1993.1022>
- [8] Finstad, K. (2010, 09). The usability metric for user experience. *Interacting with Computers*, 22, 323-327. doi: 10.1016/j.intcom.2010.04.004
- [9] Hakimi, E. (2019, Jan). *Low code development platform*. Retrieved from <https://www.slideshare.net/EhsanHakimi/low-code-development-platform>
- [10] Hecht, L. E. (2021, Dec). *Low-code platform adoption gets a boost from digital transformation*. Retrieved from <https://thenewstack.io/low-code-platform-adoption-gets-a-boost-from-digital-transformation/>
- [11] Hedau, S. R., & Mandge, O. (2021, Jun). *Low code amp; no code software development of the future*. International Research Journal of Modernization in Engineering Technology and Science, 03(06), 1131–1135.
- [12] Jain, A., Iijima, K., Leow, A., Wong, J., & Vincent, P. (2021, Sep). Retrieved from <https://www.gartner.com/document/4005973>
- [13] Joshi, A., Kale, S., Chandel, S., & Pal, D. (2015, 01). Likert scale: Explored and explained. *British Journal of Applied Science Technology*, 7, 396-403. doi: 10.9734/BJAST/2015/14975
- [14] Lewis, J., Utesch, B., & Maher, D. (2013, 04). *Umux-lite: when there's no time for the sus*. In (p. 2099-2102). doi: 10.1145/2470654.2481287
- [15] Low-code development platform. (2022). Retrieved from <https://trends.google.com/trends/explore?date=today%205-y&q=%2Fg%2F11c6cx4nrr>
- [16] Luo, Y., Liang, P., Wang, C., Shahin, M., & Zhan, J. (2021, oct). Characteristics and challenges of low-code development. In *Proceedings of the 15th ACM / IEEE international symposium on empirical software engineering and measurement (ESEM)*. ACM. doi: 10.1145/3475716.3475782
- [17] OutSystems. (2019). *Application development trends 2019 - global survey report*. Retrieved from <https://www.outsystems.com/1/state-app-development-trends/>
- [18] Pantelimon, S.-G., Rogojanu, T., Braileanu, A., Stanciu, V.-D., & Dobre, C. (2019). Towards a seamless integration of iot devices with iot platforms using a low-code approach. *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. doi: 10.1109/wf-iot.2019.8767313
- [19] Ploder, C., Bernsteiner, R., Schlögl, S., & Gschliesser, C. (2019). The future use of lowcode/nocode platforms by knowledge workers – an acceptance study. *Communications in Computer and Information Science*, 445–454. doi: 10.1007/978-3-030-21451-7_38
- [20] Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020, 08). Supporting the understanding and comparison of low-code development platforms. doi: 10.1109/SEAA51224.2020.00036
- [21] Sanchis, R., García-Perales, , Fraile, F., & Poler, R. (2020). Low-code as enabler of digital transformation in manufacturing industry. *Applied Sciences*, 10(1). Retrieved from <https://www.mdpi.com/2076-3417/10/1/12>
- [22] Vincent, P., Driver, M., & Wong, J. (2019, Feb). *Low-code development technologies evaluation guide*. Gartner Inc. Retrieved from <https://www.gartner.com/en/documents/3902331/low-code-development-technologies-evaluation-guide>
- [23] Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, 52(10), 376-381. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2405896319309152> (13th IFAC Workshop on Intelligent Manufacturing Systems IMS 2019) doi: <https://doi.org/10.1016/j.ifacol.2019.10.060>
- [24] White, S. A. (2004). *Introduction to bpmn*. Ibm Cooperation, 2(0), 0.
- [25] Wong, J., Iijima, K., Leow, A., Jain, A., & Vincent, P. (2021, Sep). *Magic quadrant for enterprise low-code application platforms*. Retrieved from <https://www.gartner.com/document/4005939>

■

Patrik Rek je asistent na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Leta 2019 je magistriral iz tematike analize kompleksnosti uporabe orodja Vortex za ustvarjanje celovitih rešitev decentraliziranih aplikacij. Njegovo raziskovalno delo obsega domeno tehnologij veriženja blokov in decentraliziranih aplikacij, še posebej na področju programskega inženirstva. Ukvarja se tudi z raziskovanjem na področju razvoja aplikacij z malo ali nič programiranja. Trenutno je doktorski študent na bolonjskem doktorskem študijskem programu Računalništvo in informatika.

■

Tina Beranič je docentka 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) in leta 2018 pridobila naziv Preizkušen revizor informacijskih sistemov .

■

Marjan Heričko je redni profesor za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, kjer je nosilec več predmetov, ki so v pristojnosti Inštituta za informatiko. Je predstojnik Inštituta za informatiko ter vodja Laboratorija za informacijske sisteme. Doktoriral je leta 1998 na Univerzi v Mariboru na področju zagotavljanja kakovosti objektno orientiranega razvoja programske opreme. Njegovo raziskovalno delo zajema vsa področja razvoja sodobnih informacijskih rešitev in storitev s poudarkom na naprednih pristopih k modeliranju in načrtovanju informacijskih sistemov, načrtovalskih vzorcih in metrikah.

Ena igra za vse: Generiranje personalizirane vsebine v mobilnih igrah

Davor Hafnar, Jure Demšar

University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, Ljubljana, Slovenia
dh2335@student.uni-lj.si, jure.demsar@fri.uni-lj.si

Izvleček

Proceduralno generiranje vsebine uporablja razne algoritme za ustvarjanje velikih količin novih vsebin v video igrah in s tem znižuje stroške produkcije. Tako generirana vsebina pa je navadno enaka za vse igralce – ni prilagojena in optimizirana za posameznega igralca. Jedro našega raziskovanja je zato usmerjeno v izboljšanje proceduralnega generiranja vsebine, tako da to upošteva karakteristike igralca. Izboljšavo nameravamo doseči z uporabo sodobnih tehnik strojnega učenja, s pomočjo katerih lahko izluščimo karakteristike posameznega igralca. Te karakteristike bodo potem uporabljene kot vhodni parametri algoritma za produkcijo personalizirane vsebine. Pričakujemo, da bo tovrstna personalizacija proceduralnega generiranja vsebine imela pozitiven vpliv na igralčevo izkušnjo.

Ključne besede: mobilne igre, proceduralno generiranje vsebine, personalizacija

ONE GAME FITS ALL: PERSONALIZED CONTENT GENERATION IN MOBILE GAMES

Abstract

Procedural content generation uses algorithmic techniques to create large amounts of new content for games and thus reduces the cost of production. However, this content generation is typically the same for all players and is not used to personalize and optimize the game for players' characteristics. Thus, the core of our research is the improvement of procedural content generation through personalization. We plan to achieve personalization by using modern machine learning algorithms to learn the characteristics of the player. These characteristics will be then used as input parameters for procedural content generation algorithms to produce personalized content. We expect that personalized procedural content generation will have a positive effect on the user's gameplay experience.

Keywords: Mobile gaming, Procedural content generation, Personalization

1 INTRODUCTION

To succeed in a highly saturated mobile game market just creating a good game is not enough. You usually also need a sizeable advertising investment. Because of this, the cost per install can be high, and retaining players is very important. We can keep the player in the game by providing new and engaging content, which can, again, be expensive, as it requires a lot of

time from developers, artists, and game designers. A popular way to address this is the procedural content generation (PCG).

PCG uses algorithmic techniques to create content for games. It is employed to increase replay value, reduce production costs and effort, or save storage space. Apart from accounting for difficulty, it is usually not personalized – content is not generated

to match the preferences of a specific player. In other words, procedurally generated content is the same for all players, regardless of their play style.

Our idea is to improve PCG algorithms by empowering them with modern ML (machine learning) algorithms. Some interesting work has already been done in the area, like the proposal of the Play Data Profiling (PDP) framework [1], which included psychophysiological measurements and eye tracking for the purpose of adjusting the game to the given player's current state. There has also been some industry-driven research, mostly done by game studios [2], whose data science teams aim to improve the KPIs (key performance indicators) of their games. Only a limited amount of their research appears to be published and even in those cases, implementation details are very scarce.

To address this challenge, we plan to develop a proof-of-concept framework for personalized PCG in mobile games. The framework will be an end-to-end solution for personalized PCG. To offer a consistent experience for all players we want our approach to be able to generate decent personalized levels from the beginning. For this, we need to address the cold start problem. Alongside we plan to develop a prototype that we will also use to compare our approach to existing game design approaches to validate the benefits it brings to players.

2 RELATED WORK

A recent article on the state of AI research on personalized games confirms the market need for our proposed solution. Zhu and Ontañón [3] argue that the research can benefit from more player-centred perspectives. Authors argue that computer games represent an ideal research domain for the next generation of personalized digital applications and that to reach the full potential of personalized games, a player-centred approach is necessary.

Personalization. A personalized approach aimed at player retention was utilized by Milošević et al. [4]. They focused on retaining players in a mobile game by utilizing early churn prediction and personalized player targeting. They first predicted which players are likely to churn and sent each one of them a personalized notification. They determined that such a personalized approach can retain players that would have otherwise left the game.

An interesting model for personalization was proposed by researchers Rajanen and Rajanen [1]. Their

idea is that a gaming system should be built in a way that it collects real-time play data which is used for player profiling even after the system has been developed. They introduce a PDP model which proposes that gamification elements are adapted based on the data derived from the interaction and the personal data of the player. A periodic reassessment of the player may determine that a player is moved from one profile cluster to another.

Procedural content generation. Content generation is currently a very active field for ML research. There generated content can vary from visual to audio and narrative aspects of the game, according to Karpouzis and Tsatiris [5]. Especially popular are fully automated PCG techniques, which are usually matched with a generation algorithm and relevant constraints in order for the game to make sense.

Researchers at Electronic Arts [2] tackled dynamic difficulty adjustment as an optimization problem. The match 3 game generated random levels with varying difficulty. The goal of ML-based optimization in their case was to maximize player engagement over the entire game. Their solution resulted in an increase in core engagement metrics such as rounds played and gameplay duration, however, it only focused on adjusting the difficulty, while our research aims to facilitate broader personalization.

Synthetic player data. A different area where ML can improve in mobile games is testing the game's playability. Gudmundsson et al. [6], researchers from game developer King, tested human-like playtesting in computer games based on ML trained on player data. The model was used to predict the difficulty of newly designed levels without requiring real players, which drastically reduced test times. Another solution was proposed by Shin et al. [7]. To automate playtesting in a match 3 type game, they used an approach based on reinforcement learning, which produced performance within a 5 per cent margin of human performance.

3 METHODOLOGY

For the purpose of our research, we developed and published a basic match 3 mobile game. The game is developed in C# programming language using the Unity game engine and will interact with Google Cloud using Google's Firebase platform. Products from the Google Cloud such as Firebase, BigQuery and Cloud Functions were used for data collection

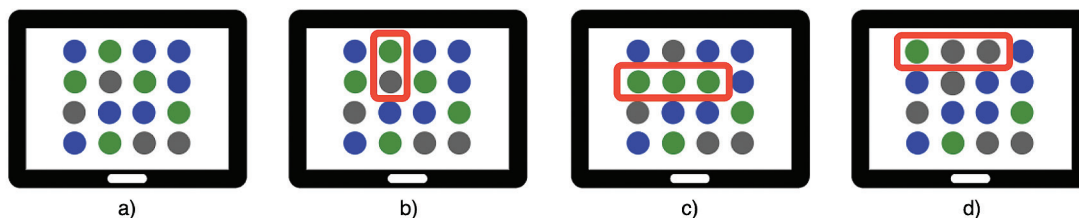


Figure 1: (a) A basic illustration of game mechanics in our match 3 puzzle game. (b) The player's objective is to find and match three board elements of the same colour. He can do that by swapping neighbouring pieces. (c) Once three or more elements of the same colour form a line, the player is awarded a certain amount of points, while the matching elements are removed from the board. (d) To fill the void, elements above the removed line are moved down and replaced by newly assigned elements.

and processing as well as ML training and inference. To train the agents, we plan to develop an algorithm based on Proximal Policy Optimization (PPO) [8] which belongs to the family of deep reinforcement learning algorithms.

Our game uses match 3 mechanics. The basic gameplay is visualized in Figure 1. For simplicity, our game consists of a sequence of levels without a hub that connects them. Before starting, the user is shown a pop-up with objectives and limitations for the level. An example of an objective is a score that the player has to reach combined with the number of elements of specific colour he needs to connect and consequently remove from the board. The player is constrained by the number of moves. The time limit can also be used to create additional pressure.

The popularity of Unity, the game engine used for development, as well as the popularity of the match 3 genre help with development as multiple useful templates, are readily available. Our game will be based on one such template.

4 RESULTS

Before we could evaluate our approach, we had to prepare a prototype mobile game and connect it to a backend system which collected the data. Initial tests, therefore, do not test our fully developed solution but serve as an approximation by comparing a personalized and static solution. The design of our game allows for concurrent testing with any number of players. However, for the initial test, where we also wanted to make sure everything works as expected, we decided on a smaller group of fifteen testers personally asked to play.

The basic personalization utilized assigned the user to be either in a control group that is served normal levels or is in the test group that is, based on the player's performance in the diagnostic level, served easy, normal, tactical or dynamic levels. Seven users

out of fifteen were randomly assigned to a group that got served levels using the basic personalization. Two of those were assigned tactical, one was assigned easy, and four were assigned normal types of levels. We asked the users to rate levels. Scores ranged from 1 to 5, the latter representing the highest level of satisfaction. To the experimental group, we served the levels for the user based on how they played the first, diagnostic level. To the control group, levels were served the same for all users.

Initial results, as shown in Table 1, tell us that basic personalization will not necessarily outperform conventional game design. A more advanced approach is required to improve performance. In our initial testing, the average rating of levels by groups that were served either static or personalized levels was almost identical. The personalization in the test did not rely on ML, but on simple, hardcoded conditions. The results are about in line with expectations, as personalization that would improve the results is likely to require more sophistication.

5 CONCLUSION

In this paper, we presented a proof-of-concept framework for personalized PCG in mobile games. The framework will be an end-to-end solution for personalized PCG. To offer a consistent experience for all players we want our approach to be able to gene-

Tabela 1: Average scores assigned to individual levels by the two user groups were very similar. The p-value calculated using an independent two-tailed t-test was 0.87, way above the commonly used 0.05 mark. Based on the results of the test we can conclude that there is no statistically significant evidence of a difference between the two groups.

	Players	Average level score	Standard deviation
Static levels	8	3,89	1,12
Personalized levels	7	3,85	0,98

rate decent personalized levels from the beginning. A direct result of the approach is also that players are provided with an unlimited, personalized game experience. While there have been developments in real-time content generation using ML, procedurally generated content is typically the same for all players, regardless of their play style.

The weakness of our approach is that, as the initial results show, the benefits are not guaranteed and a lot of experimentation is likely to be required to realize an improvement. Our approach also required a complex software solution before we could focus our work on the algorithms.

Our framework will make it easier for game developers to switch from conventional game design to a design that – by leveraging modern ML approaches – increases their chance of creating a game that the players will enjoy. Our research aims to bridge the gap between purely academic research, which is often not directly usable by industry, and industry research, which has practical goals but is not easily reproducible by studios with fewer resources.

6 REFERENCES

- [1] M. Rajanen, D. Rajanen, "Personalized Gamification: A Model for Play Data Profiling," *Proceedings of the First International Workshop on Data-Driven Gamification Design (DDGD 2017)*, 2017
- [2] S. Xue, M. Wu, J. Kolen, N. Aghdaie in K. A. Zaman, "Dynamic difficulty adjustment for maximized engagement in digital games," *v 26th International World Wide Web Conference 2017, WWW 2017 Companion*, pp. 465–47, 2019
- [3] J. Zhu and S. Ontañón, "Player-Centered AI for Automatic Game Personalization: Open Problems," *ACM Int. Conf. Proceeding Ser.*, 2020
- [4] M. Milošević, N. Živić, and I. Andjelković, "Early churn prediction with personalized targeting in mobile social games," *Expert Syst. Appl.*, vol. 83, pp. 326–332, 2017
- [5] K. Karpouzis, G. Tsatiris, "AI in (and for) Games," *ArXiv*, <https://arxiv.org/abs/2105.03123>, 2021
- [6] S. F. Gudmundsson, P. Eisen, E. Poromaa, A. Nodet, S. Purmonen, B. Kozakowski, R. Meurling and L. Cao, "Human-Like Playtesting with Deep Learning," *IEEE Conference on Computational Intelligence and Games, CIG, 2018-August*, 2018
- [7] Y. Shin, J. Kim, K. Jin, , Y. Kim, "Playtesting in Match 3 Game Using Strategic Plays via Reinforcement Learning," *IEEE Access*, vol. 8, pp. 51593–51600, 2020
- [8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford O. Klimov, "Proximal policy optimization algorithms." *arXiv preprint 1707.06347.*, 2017

■

Davor Hafnar is a doctoral student at the Faculty of Computer and Information Science, University of Ljubljana. He obtained his master's degree from the University in Ljubljana in 2020. His research focus is on leveraging machine learning to solve real-life problems.

■

Jure Demšar is an Assistant Professor at the Faculty of Computer and Information Science, University of Ljubljana. His main areas of research are neuroimaging, data analysis, collective behaviour, and game development. He obtained his PhD degree in Computer Science from University of Ljubljana in 2017.

■ Ocenjevanje kakovosti prstnih sledi z ansambli globokega učenja

Tim Oblak, Peter Peer

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana
{tim.oblak,peter.peer}@fri.uni-lj.si

Izvleček

Ocena kakovosti je pomemben korak za identifikacijo prstnih sledi s kraja zločina. Pogosto se izvaja v okviru forenzične preiskave, izvajajo pa ga usposobljeni preiskovalci in je ponavadi precej subjektiven. Cilj našega dela je razviti avtomatizirano metodo ocenjevanja kakovosti prstnih sledi, ki bi pomagala preiskovalcem pri njihovem delu. V tem delu sodobne tehnike globokega učenja prenesemo na področje ocenjevanja kakovosti prstnih sledi, ovrednotimo prednosti in slabosti te metodologije ter načrtujemo smer nadaljnjega razvoja na področju. Predlagamo novo metriko kakovosti, ki s pomočjo posebne metode fuzije združuje posamezne napovedi ansambla globokih modelov. Predlagani pristop zagotavlja izboljšano zmogljivost napovedi, hkrati pa skrajša čas obdelave za vsaj 15-krat v primerjavi z obstoječimi najsodobnejšimi rešitvami.

Ključne besede: forenzika, globoko učenje, ocena kakovosti, prstne sledi

FINGERMARK QUALITY ASSESSMENT WITH DEEP LEARNING ENSEMBLES

Abstract

Quality assessment is an important step when trying to identify fingerprints from a crime scene. Often done in the scope of forensic investigation, it is performed by trained examiners and tends to be rather subjective. The goal of our work is to develop an automated fingerprint quality assessment method which would assist the examiners in their work. In this paper, we introduce modern deep learning techniques into the field of fingerprint quality assessment, evaluate the advantages and disadvantages of this methodology, and identify the key aspects for further development in the field. We propose a new quality metric which works by fusing individual predictions of an ensemble of deep models. The proposed approach provides improved prediction performance while reducing processing time by at least a factor of 15 compared to existing solutions.

Keywords: Forensics, deep learning, quality assessment, fingerprints

1 UVOD

Prstne sledi (latentni prstni odtisi) so posebna vrsta odtisov kože s konic naših prstov, ki jih najdemo v nenadzorovanih okoljih, na primer v okviru forenzične preiskave [1]. V takih primerih odtis kožnega vzorca ni nadzorovan in med postopkom pride do napak. Med analizo sledi preiskovalci najprej ocenijo

njihovo kakovost. Ta vrednost pomeni, (i) kako uporabna je sled kot forenzični dokaz, (ii) določa, kakšen bo postopek obdelave sledi, in (iii) kaže na uspešnost potencialnega ujemanja s referenčnimi prstnimi odtisi. Ocena kakovosti prstnih sledi ima torej velik vpliv na preiskavo. Kot je značilno za človeško naravo, so tudi odločitve forenzičnih preiskovalcev zelo subjek-

tivne [7, 8, 12]. Uporabna prstna sled se zato lahko zavrže že zgodaj v preiskavi ali pa se preveč časa nameni prepoznavanju prstnih sledi slabe kakovosti. Da bi zmanjšali to subjektivnost, želimo avtomatizirati proces ocenjevanja kakovosti prstnih sledi.

Trenutno več dejavnikov omejuje razvoj novih metod avtomatiziranega ocenjevanja kakovosti prstnih sledi (angl. automated fingerprint quality assessment – AFQA). Komercialni ponudniki svoje algoritme ščitijo kot intelektualno lastnino. Nekatere metode so bile razvite posebej za organe pregona in niso javno dostopne [9]. Običajno raziskave vključujejo nadzorovano učenje napovedovalnih modelov [2, 3, 4], redko pa avtorji objavijo tudi kodo, naučene modele, podatke ali referenčne vrednosti. Čeprav obstajajo odprtokodne rešitve za ocenjevanje kakovosti odtisov s prstov [11], so te večinoma osredotočene na ocenjevanje optično branih prstnih odtisov v nadzorovanem okolju in niso primerne za uporabo v forenziki. Druga vrsta objav [13, 10] se opira na ročno sestavljene značilke, zaradi česar so metode razumljive, vendar je količina izluščenih informacij omejena. Zaradi teh dejavnikov smo ugotovili, da bi bila nova metoda ocenjevanja kakovosti prstnih sledi koristna za to znanstveno področje.

Doslej metode globokega učenja na tem področju še niso bile uporabljene. V tem delu preučujemo, kako je mogoče konvolucijske nevronske mreže (angl. convolutional neural network – CNN) najbolje uporabiti v kontekstu metod AFQA in predlagamo novo metodo, prikazano na sliki 1. V okviru raziskave (i) zgradimo ansambel CNN-jev, ki so naučeni na različnih množicah oznak kakovosti. Za označevanje našega nabora podatkov uporabljamo obstoječe komercialne in zaprtokodne metode AFQA. V drugem delu (ii) predlagamo metodo fuzije, ki vzame posamezne glasove ansambla in jih združi, da ustvari enotno, robustnejšo vrednost napovedane kakovosti.

2 ANSAMBEL ZA OCENJEVANJE KAKOVOSTI PRSTNIH SLEDI

Formulacija problema. Za napovedovanje vrednosti kakovosti y iz vhodne slike x v tem delu načrtujemo strategijo učenja, kjer naučeni modeli F_{CNN} napovejo ocene kakovosti, ki so čim bližje resničnim oznakam: $\hat{y} = F_{CNN}(x; \theta_{CNN})$, kjer je \hat{y} izhod konvolucijske nevronske mreže, θ_{CNN} pa so učljive uteži mreže. V namen ansambelskega učenja predlagamo arhitekturo modela F_{CNN} posplošimo tako, da izhod $\hat{y} \in \mathbb{R}^m$ definiramo kot vektor kakovosti velikosti $m \in \mathbb{N}$. V konfiguraciji ansambla je lahko F_{CNN} bodisi CNN z izhodom velikosti m ali zbirka m CNN modelov, od katerih je vsak naučen na svojem naboru oznak kakovosti.

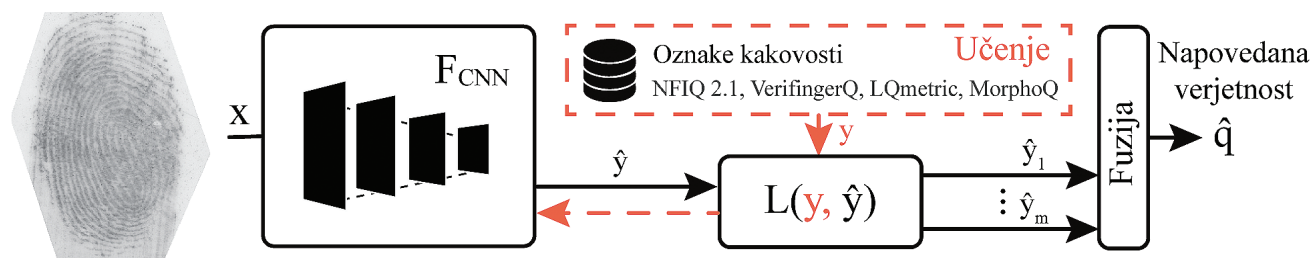
Optimalne parametre θ iščemo tako, da minimiziramo funkcijo izgube $\mathcal{L}(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$, kjer je Y nabor resničnih vektorjev kakovosti, \hat{Y} je množica napovedanih vektorjev kakovosti, n pa je število slik prstnih sledi v učni množici.

Arhitektura CNN.

Kot osnovno ogrodje našega modela lahko uporabimo katero koli od uveljavljenih arhitektur CNN. V kontekstu prstnih oznak nam razširitev zaznavnega polja v CNN omogoča obdelavo slik prstnih sledi, ne da bi vnaprej poznali ločljivost slik. Ker je ciljna vrednost zvezno število, spremenimo izhod mreže in ustvarimo regresijski model. Za povečanje zmogljivosti mreže na podlagi predhodnega testiranja na koncu mreže dodamo še en popolnoma povezan sloj s 512 nevroni. Za tem dopolnimo mrežo še z izhodno plastjo z m nevroni.

Metoda fuzije.

Da končno napoved kakovosti naredimo bolj robustno in čim bolj pokrijemo širok razpon možnih lastnosti prstnih sledi, lahko napovedi ansambla zdru-



Slika 1: Model za ocenjevanje kakovosti prstnih sledi. Metoda, predlagana v tem delu, je sestavljena iz CNN ansambla in metode fuzije, ki posamezne glasove združi v končno vrednost kakovosti.

žimo s pomočjo fuzije. Za pravilno uteževanje posameznih ocen kakovosti želimo izračunati projekcijo $R^n \rightarrow R$ iz m-dimenzionalnega vektorja ocen kakovosti y na enodimenzionalno končno vrednosti q . S pomočjo PCA najprej najdemo os največje variance znotraj m-dimenzionalnega prostora glasov ansambla, potem pa posamezne vektorje kakovosti preslikamo na to enodimenzionalno os. Preslikavo definiramo na naslednji način:

$$\hat{q}_{pca} = W (\hat{y} - \bar{y}), \quad (1)$$

kjer je $W \in R^{1 \times n}$ lastni vektor največje glavne komponente (angl. principal component), pridobljen z metodo PCA, \hat{y} je napovedani vektor kakovosti od katerega odštejemo povprečni vektor resničnih vrednosti kakovosti \bar{y} , $\hat{q}_{pca} \in R$ pa je prva glavna komponenta. Za transformacijo končne vrednosti v razpon kakovosti (0, 100) izhod dodatno normaliziramo:

$$\hat{q} = (\hat{q}_{pca} - q_{pca}^{min}) / (q_{pca}^{max} - q_{pca}^{min}), \quad (2)$$

kjer sta q_{pca}^{max} največja, q_{pca}^{min} pa najmanjša glavna komponenta, izračunana z resničnih vrednosti učne množice, \hat{q} pa je končni rezultat fuzije napovedanih vrednosti kakovosti.

3 REZULTATI

Ekperimentalno okolje.

Za učenje modelov uporabljamo dve množici prstnih sledi, NIST SD 302 [5] in NIST SD 301 [6]. Obe vsebujeta prstne sledi, ki so jih z različnih površin pridobili usposobljeni forenzični preiskovalci v simuliranem okolju. Podatke smo razdelili v učno množico (10.000 prstnih sledi + 2000 prstnih odtisov z množice SD 302) in testno množico (1200 prstnih sledi z množice SD 301). Namesto ročnega označevanja resničnih

vrednosti kakovosti smo uporabili obstoječe algoritme za ocenjevanje kakovosti in tako ustvarili ciljne vrednosti za učenje. Uporabili smo štiri metode; NFIQ 2.1 [11], VerifingerQ, MorphoQ in LQmetric [9]. Za vrednotenje uspešnosti naučenih modelov smo uporabili metrike, ki se v literaturi običajno uporabljajo za regresijske naloge, in sicer povprečno absolutno napako (angl. mean absolute error – MAE) in r-kvadrat (angl. r-squared - R^2).

Ovrednotenje modelov.

Naš cilj je naučiti napovedovalni model za vsako množico ciljnih vrednosti kakovosti, pridobljenih z metodami NFIQ 2.1, VerifingerQ, MorphoQ in LQmetric. Naučene modele poimenujemo NFQ, VFQ, MOR oziroma LQM. Poleg tega za vsakega od štirih modelov testiramo štiri različna ogrodja CNN-jev, ki se pogosto uporabljajo v literaturi (ResNet, DenseNet, EfficientNet in Inception). Rezultati so prikazani v tabeli 1. Največje razlike so opažene med modeli NFQ, kjer se vrednost R^2 giblje od 0,37 do 0,60. Zdi se, da so modeli, usposobljeni z drugimi vrednostmi kakovosti, bolj konsistentni med različnimi arhitekturami. Na splošno je natančnost regresije izbranih osnovnih arhitektur relativno visoka. MAE v povprečju predstavlja okoli 5 % možnega razpona kakovosti od 0 do 100. Pri ocenjevanju ogrodij je na splošno najboljši DenseNet, sledi mu EfficientNet. DenseNet dosega najboljšo korelacijsko zmogljivost za tri od štirih modelov in ima najvišje povprečno vrednost $R^2 = 0,74$. Prav tako izstopa kot najboljši napovedovalec metode NFIQ 2.1 v primerjavi z drugimi arhitekturami. Drug vidik je čas izvedbe. Za dano serijo slik naučen model napove kakovosti v konstantnem času. Od vseh štirih ogrodij sta EfficientNet in Inception-v3 očitno najhitrejša, saj oba obdelata serijo slik v približno 10 ms. V primerjavi NFIQ 2.1 porabi najmanj 150 ms za obdelavo slike prstnega odtisa.

Tabela 1: Rezultati vrednotenja napovednikov CNN. Modeli so testirani na testni množici in primerjani z uporabo metrike MAE in R^2 . Najboljšo napovedno zmogljivost je ansambel dosegel z uporabo ogrodja DenseNet-121.

Model	ResNet-50		DenseNet-121		EfficientNet-b0		Inception-v3	
	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
NFQ	3,41	0,49	3,03	0,60	3,14	0,55	3,38	0,37
VFQ	4,29	0,73	4,20	0,76	4,20	0,75	4,12	0,75
LQM	7,8	0,79	7,44	0,79	6,92	0,83	7,64	0,78
MOR	2,11	0,81	2,01	0,83	1,99	0,82	1,95	0,82
Čas izvajanja (ms)	19,59 ± 0,10		21,05 ± 0,85		11,1 ± 1,89		10,2 ± 0,89	

CNN modeli so tako vsaj 15-krat hitrejši v primerjavi s konvencionalno metodo z ročnim luščenjem značilnk. Sami po sebi pa CNN-ji ne nudijo načina za razlago svojih odločitev. Z vidika razložljivosti ostaja klasičen pristop ročno izbranih značilnk boljša izbira. Več raziskav je potrebnih za združitev učinkovitosti CNN-jev in interpretabilnosti metod z ročno izbranimi značilnkami.

Ovrednotenje fuzije.

Da bi ocenili, kako se rezultat fuzije obnese v primerjavi z ostalimi metodami, lahko izmerimo njegovo napovedno moč. Na sliki 2 smo vzeli naš najuspešnejši model (DenseNet-121) in prikazali krivulje ujemanja za celoten ansambel, vključno z novo metofo fuzije. Najprej definiramo idealen scenarij, kjer krivulja sledi diagonalni pikčasti črti. V našem primeru želimo, da napovedana metrika kakovosti korelira z verjetnostjo, da bomo za dano prstno sled našli ujemajoč se referenčni prstni odtis na prvem mestu (rank-1). Če je idealna krivulja diagonalna črta, potem kakovost 20 pomeni, da obstaja 20 % verjetnost rank-1 identifikacije. S krivulj slike 2 je razvidno, da imata modela NFQ in MOR najvišje naraščajoče krivulje, ki dosežejo 100 % verjetnost že okoli kakovosti 40. Na drugi strani je krivulja modela LQM vedno pod diagonalno črto, kar pomeni, da model je preveč radodaren z visokimi ocenami kakovosti in da rank-1 ujemanja niso zagotovljena, tudi če je ocena kakovosti dobra. Idealni situaciji se najbolje prilaga predlagana metoda fuzije, sledi pa ji model VFQ. Glede na to, da metoda fuzije upošteva tudi poraz-

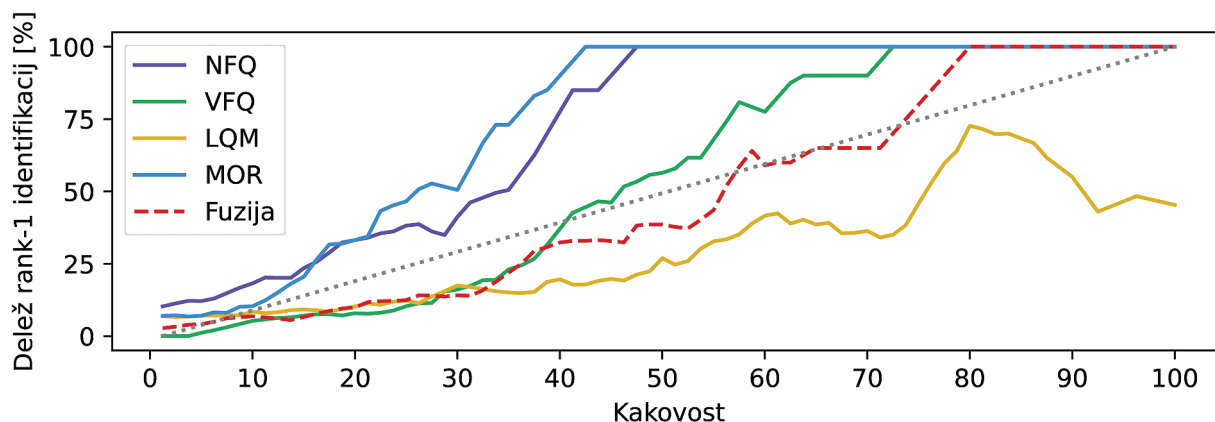
delitve prvotnih oznak v učni množici, je ta sposobna bolje utežiti prispevke posameznih glasov ansambla in zato deluje bolj robustno v nasprotju s posameznimi napovedmi.

4 ZAKLJUČEK

V tem delu smo predlagali nov pristop za ocenjevanje kakovosti prstnih sledi s pomočjo globokega učenja. Metoda je sestavljena iz ansambla modelov CNN in metode za fuzijo, ki posamezne napovedi ansambla združi v eno samo robustno vrednost kakovosti. S tem delom forenzični in biometrični skupnosti nudimo sodobno, zanesljivo in odprtokodno metriko kakovosti prstnih sledi. V prihodnosti bomo nadaljevali z uporabo metod globokega učenja, vendar pa se bomo osredotočili na razložljivost odločitev modela in pa celovito predstavitev končnih rezultatov.

LITERATURA

- [1] Jeffery G. Barnes. History. In Fingerprint sourcebook, chapter 1, pages 5–22. U.S. Department of Justice, National Institute of Justice, 2010.
- [2] Kai Cao, Tarang Chugh, Jiayu Zhou, Elham Tabassi, and Anil K Jain. Automatic latent value determination. In International Conference on Biometrics, pages 1–8. IEEE, 2016.
- [3] Tarang Chugh, Kai Cao, Jiayu Zhou, Elham Tabassi, and Anil K. Jain. Latent Fingerprint Value Prediction: Crowd-Based Learning. IEEE Transactions on Information Forensics and Security, 13(1):20–34, 2018.
- [4] Jude Ezeobiesi and Bir Bhanu. Latent fingerprint image quality assessment using deep learning. In Conference on Computer Vision and Pattern Recognition Workshops, pages 508–516. IEEE, 2018.
- [5] Gregory Fiumara, Patricia Flanagan, John Grantham, Kenneth Ko, Karen Marshall, Matthew Schwarz, Elham Tabassi, Brian Woodgate, and Christopher Boehnen. NIST Special Database 302: Nail to Nail Fingerprint Challenge. Technical



Slika 2: Krivulje rank-1 identifikacije. V tem grafu je razvidno razmerje med napovedano kakovostjo različnih metod in deležom rank-1 identifikacij. Za vsako kakovost v razponu od 0 do 100 prikazemo delež prstnih odtisov te kakovosti, ki so bili uspešno najdeni na prvem mestu s pomočjo avtomatizirane metode ujemanja prstnih sledi. Rezultati združenega ansambla imajo najboljšo napovedno moč v primerjavi z ostalimi metodami.

- Report 2007, NIST, 2018.
- [6] Gregory Fiumara, Patricia Flanagan, Matthew Schwarz, Elham Tabassi, and Christopher Boehnen. Nist special database 301: Nail to nail fingerprint challenge dry run. Technical Report 2002, NIST, 2018.
- [7] R Haraksim, J Galbally, and L Beslay. Study on Fingerprint and Palmmark Identification Technologies for their Implementation in the Schengen Information System. EUR 29755 EN, Publications Office of the European Union, 2019.
- [8] R Austin Hicklin, Joann Buscaglia, Maria Antonia Roberts, Stephen B Meagher, William Fellner, Mark J Burge, Matthew Monaco, Davill Vera, Larry R Pantzer, Calvin C Yeung, and Ted N Unnikumaran. Latent Fingerprint Quality: A Survey of Examiners. *Journal of Forensic Identification*, 61(4):385–419, 2011.
- [9] Nathan D Kalka, Michael Beachler, and R Austin Hicklin. LQMetric: A Latent Fingerprint Quality Metric for Predicting AFIS Performance and Assessing the Value of Latent Fingerprints. *Journal of Forensic Identification*, 70:443–463, 2020.
- [10] Anush Sankaran, Mayank Vatsa, and Richa Singh. Automated clarity and quality assessment for latent fingerprints. *International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, 2013.
- [11] Elham Tabassi, Martin Olsen, Oliver Bausinger, Christoph Busch, Andrew Figlarz, Gregory Fiumara, Olaf Henniger, Johannes Merkle, Timo Ruhland, Christopher Schiel, and Michael Schwaiger. NIST Fingerprint Image Quality 2, NISTIR 8382. National Institute of Standards and Technology, 2021.
- [12] Bradford T. Ulery, R. Austin Hicklin, Jo Ann Buscaglia, and Maria Antonia Roberts. Repeatability and reproducibility of decisions by latent fingerprint examiners. *PLoS ONE*, 7(3), 2012.
- [13] Soweon Yoon, Eryun Liu, and Anil K Jain. On latent fingerprint image quality. In *Computational Forensics*, pages 67–82. Springer, 2012.

■

Tim Oblak je doktorski študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Leta 2020 je zaključil dvojni magisterij na Univerzi v Ljubljani in Tehnični univerzi v Gradcu. V svojih raziskavah uporablja metode računalniškega vida s poudarkom na globokem učenju. V preteklosti se je ukvarjal z rekonstrukcijo 3D prostora, trenutno pa v sklopu doktorskega študija razvija nove metode na področju digitalne forenzike in identifikacije prstnih sledi.

■

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. 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. 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.

▼ Vpliv kognitivne obremenjenosti na vedenjske vzorce posameznikov pri overjanju uporabnikov

Andraž Krašovec, Veljko Pejović

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana
ak6688@student.uni-lj.si, veljko.pejovic@fri.uni-lj.si

Izveček

Gesla in biometrični pristopi, na primer prstni odtisi, prevladujejo v svetu overjanja uporabnikov. Zaradi pomislekov glede varnosti, zasebnosti in priročnosti, znanost išče ustrezne nadomestke, vključujoč metode na podlagi vedenjskih vzorcev posameznikov. Kljub temu, da ti pristopi odpravijo večino obstoječih pomanj-kljivosti, zaenkrat še niso dovolj natančni za splošno rabo. Glavni razlog za to je dejstvo, da uporabnikovo vedenje ni nespremenljivo. En izmed potencialnih vzrokov sprememb v obnašanju je spremenljiva kognitivna obremenjenost posameznika pri izvajanju različnih nalog. V tem delu na podlagi podatkov devetih prostovoljcev predstavimo raziskavo vpliva kognitivne obremenjenosti na vedenjske vzorce z analizo EEG in senzorskih podatkov. S pomočjo algoritma naključnega gozda razlikujemo med štirimi različnimi stopnjami kognitivne obremenjenosti. Dosežemo 89,79% in 76% natančnost na podlagi EEG in senzorskih podatkov, s čimer prikažemo povezavo med podatkovnima zbirkama.

Ključne besede: Kognitivna obremenjenost, overjanje uporabnikov, vedenjski vzorci.

EFFECT OF COGNITIVE LOAD ON BEHAVIOURAL PATTERNS OF INDIVIDUALS FOR USER AUTHENTICATION

Abstract

Passwords and physical biometric approaches, such as fingerprints, are prevalent in the field of user authentication. Security, privacy, and convenience concerns gave rise to alternative techniques, including methods based on behavioural patterns of users. Despite mitigating many existing issues, the performance of such authentication approaches is not yet up to par with more traditional methods. One factor that potentially reduces the performance of behaviour-based systems is the level of cognitive load a person is under. This paper presents a preliminary study on the potential of EEG data for inference of cognitive load and the effect cognitive load has on the behavioural patterns of users. We gather and process EEG and behavioural data and apply a Random Forest algorithm to it to predict four different levels of cognitive load. We achieve an 89.79% and 76.00% accuracy from EEG and behavioural patterns data respectively, displaying the connection between the two.

Keywords: Cognitive load, user authentication, behavioural Patterns

1 UVOD

Gesla so že desetletja najbolj pogost način overjanja uporabnikov [16]. Šele nedavno so se geslom pridružile tudi biometrične metode, na primer prstni odtisi in prepoznavna obraza. Pristopa predstavljata pomembna koraka v zgodovini overjanja uporabni-

kov, vendar ne zagotavljata sistema, ki bi bil varen in priročen za uporabnika. Overjanje na podlagi vedenjskih vzorcev, ki jih lahko opazujemo s pomočjo senzorjev v prostoru, se zanaša na uporabnikove nezavedne, vsakodnevne akcije in predstavlja zanimivo alternativo klasičnim pristopom.

Kljub priročnosti takšnega pristopa je potrebno premostiti še nekaj ovir preden lahko overjanje na podlagi vedenjskih vzorcev postavimo ob bok bolj uveljavljenim metodam. Na primer, nedavne raziskave prikazujejo spremembe v vedenjskih vzorcih uporabnikov na različnih časovnih intervalih [13]. Ta problem najbolj vpliva na pogostost zavrnitve legitimnih uporabnikov kar negativno vpliva na uporabniško izkušnjo.

Eden izmed možnih vzrokov za spremembe vedenjskih vzorcev bi lahko bila trenutna stopnja kognitivne obremenitve posameznika. Kognitivna obremenjenost predstavlja obremenitev kognitivnega sistema posameznika pri opravljanju določene naloge [19]. V tem delu bomo preučili povezavo med kognitivno obremenitvijo in vedenjskimi vzorci uporabnika, ki jih lahko zaznamo s pomočjo senzorjev v okolju interneta stvari (IoT). Različne stopnje kognitivne obremenjenosti bomo prožili s prilagajanjem težavnosti eksperimentalnih nalog, ki jih prostovoljci opravljajo v prostoru, opremljenim s senzorji. Dodatno, v eksperimentu bomo nenehno spremljali možgansko aktivnost s pomočjo brezžične elektroencefalografije (EEG) čelade in s tem zgradili bogato podatkovno zbirko, ki jo bomo po koncu raziskave javno objavili.

Zato v tem delu, kolikor nam je znano, kot prvi;

- Opravimo preliminarno analizo vpliva različnih stopenj kognitivne obremenjenosti na vedenjske vzorce ljudi, zbrane s pomočjo senzorskih podatkov, v okviru overjanja uporabnikov.
- Sestavimo podatkovno zbirko, ki vključuje večmodalne senzorske IoT podatke in podatke o možganski aktivnosti na podlagi EEG. Končno podatkovno zbirko bomo javno objavili.

2 SORODNA DELA

Sistemi overitve uporabnikov na podlagi vedenjskih vzorcev predstavljajo alternativo bolj tradicionalnim metodam. Najbolj pogosto se takšni sistemi zanašajo na pametne naprave (telefoni, zapestnice) [20, 8, 17]. Manjšinski del področja, kamor se uvršča tudi naše delo, predstavljajo dela, ki uporabljajo senzorje interneta stvari, na primer v okolje postavljeni radarji in inercijske merilne naprave [22, 7]. Pogost problem s katerim se ti sistemi srečujejo, je spremenljivost človeškega vedenja, kar negativno vpliva na natančnost in posledično uporabnost sistemov.

Eden izmed možnih razlogov spremenljivosti človeškega vedenja je prav tako spreminjajoča se stopnja

kognitivne obremenjenosti posameznika. Kognitivna obremenjenost je definirana kot večdimenzionalni konstrukt, ki predstavlja obremenitev kognitivnega sistema posameznika pri izvajanju določene naloge [18]. Raziskave povezane s kognitivno obremenjenostjo domujejo v psihologiji [15], vendar lahko kognitivno obremenjenost analiziramo na mnogo področjih, kjer v študijah sodelujejo ljudje [5, 6] ali celo živali [4]. Barua et al. [1] povežejo kognitivno obremenjenost z zmožnostjo upravljanja avtomobila. Podobno Frosina et al. [10] raziščejo vpliv stopnje kognitivne obremenjenosti na vedenje osumljencev med zaslišanjem. Kolikor nam je znano, podobne analize vpliva kognitivne obremenjenosti na vedenjske vzorce za namen overjanja uporabnikov še ni opravil nihče.

EEG čelade so trenutno najbolj pogosto uporabljene naprave za prepoznavanje kognitivne obremenjenosti [9, 12]. Sodobni pristopi spajajo obdelavo podatkov s pomočjo hitre Fourierjeve transformacije z metodami strojnega učenja, najpogosteje z uporabo nevronske mreže [14, 2]. Poleg zgolj zaznavanja prisotnosti kognitivne obremenjenosti, so sodobni sistemi zmožni razlikovanja med različnimi stopnjami obremenitve ali celo zvezne napovedi.

Z zbranim znanjem opravimo prvo analizo vpliva kognitivne obremenjenosti na vedenje uporabnikov na področju sistemov overjanja. Zanašamo se na že opravljene analize na področju napovedovanja kognitivne obremenjenosti na podlagi podatkov z EEG naprav in pa tudi vedenjskih analiz, opravljenih na drugih raziskovalnih področjih.

3 ANALIZA KOGNITIVNE OBREMENJENOSTI

Ugotavljanje trenutne kognitivne obremenjenosti posameznika je zahtevna naloga. Trenutno najboljše metode se zanašajo na podatke, zbrane s pomočjo EEG naprav. V našem primeru sočasno zbiramo tudi vedenjske podatke, kar poveča kompleksnost našega sistema. V tem poglavju predstavimo postopek zbiranja in obdelave podatkov, katerih končni cilj je prepoznavanje različnih stopenj kognitivne obremenjenosti.

3.1 Eksperimentalne naloge

Vsak prostovoljec mora rešiti štiri naloge treh različnih težavnostnih stopenj v pisarniškem okolju. Prva naloga od uporabnika zahteva, da pretipka na zaslonu izpisano besedilo. Težavnostno stopnjo prilagajamo z izborom jezika (materin jezik, angleščina, madžarščina

Tabela 1: Ustvarjene značilke. EEG in ostali senzorski podatki so pri učenju uporabljeni ločeno.

Senzor	Ustvarjene značilke
EEG čelada	1 Hz moč signala (0-125 Hz)
Inercijske merilne naprave	1 Hz moč signala (0-100 Hz), povprečje, standardni odklon, stopnja prehoda povprečja
Senzorji sile	1 Hz moč signala (0-100 Hz), povprečje, standardni odklon, stopnja prehoda povprečja

na). Naslednji nalogi spadata v družino Elementarnih kognitivnih nalog [11], namensko zasnovanih za vzbujanje različnih stopenj kognitivne obremenjenosti; a) *Skriti vzorci*, kjer mora uporabnik poiskati vrčtanlik v eni izmed treh ponujenih slik, kjer se težavnostna stopnja spreminja s kompleksnostjo ponujenih slik in b) *Zasledovanje*, kjer uporabnik sledi črtam s katerimi poveže števila na levi in desni strani slike, kjer se težavnostna stopnja spreminja s kompleksnostjo povezovalnih črt. Zadnja naloga od uporabnika zahteva, da se sprehodi do sosednjega zaslona, kjer prebere in si zapomni izpisano število. Zatem se sprehodi nazaj do primarnega računalnika, kamor po izteku časa to število vnese. Težavnostna stopnja je izražena z različnim številom števk v številu. V poskus je vključeno tudi stanje mirovanja, kjer uporabniki strmijo v prazen računalniški zaslon in poskušajo umiriti misli.

3.2 Zbiranje podatkov

Podatke možganske aktivnosti zbiramo s pomočjo 4-kanalne in brezžične EEG čelade Explore podjetja Mentalab¹. Čelado pred poskusi udeležencu namestimo na glavo in prilagodimo elektrode tako, da je njihova impedanca nižja od 100 k Ω , kar zadostuje priporočilu proizvajalca. Frekvenca vzorčenja je nastavljena na 250 Hz. Zaradi nestabilne povezave med napravo in strežnikom, se podatki zbirajo na napravi in prenesejo na strežnik po opravljenih meritvah. Za zajem vedenjskih podatkov uporabimo naslednje senzorje;

- Štiri inercijske merilne naprave. Tri so pritrjene na vrh tipkovnice, medtem ko je četrta vgrajena v miško. Vsaka izmed naprav vsebuje triosni pospeškometer in žiroskop, kar nam omogoča tridimenzionalno zaznavanje pospeška in kotne hitrosti. Skupaj zajemamo 24 različnih signalov.

- Štirje senzorji sile. Pod miško in tipkovnico je postavljena plošča velikosti 70cm x 40cm, v vogalih katere so nameščeni senzorji sile, ki delujejo na podlagi spremembe upornosti s spreminjajočo se silo. Skupaj zajemamo 4 različne signale.

Pri analizi se zanašamo na vedenjske vzorce rokovanja uporabnika z računalnikom – inercijske merilne naprave zaznavajo ritem in hitrost pisanja na tipkovnico in način uporabe miške, medtem ko senzorji sile razpoznavajo držo pri delu z računalnikom. Analizo opravimo na podatkih devetih prostovoljcev (4 moški, 5 žensk), v povprečju starih 28,4 let.

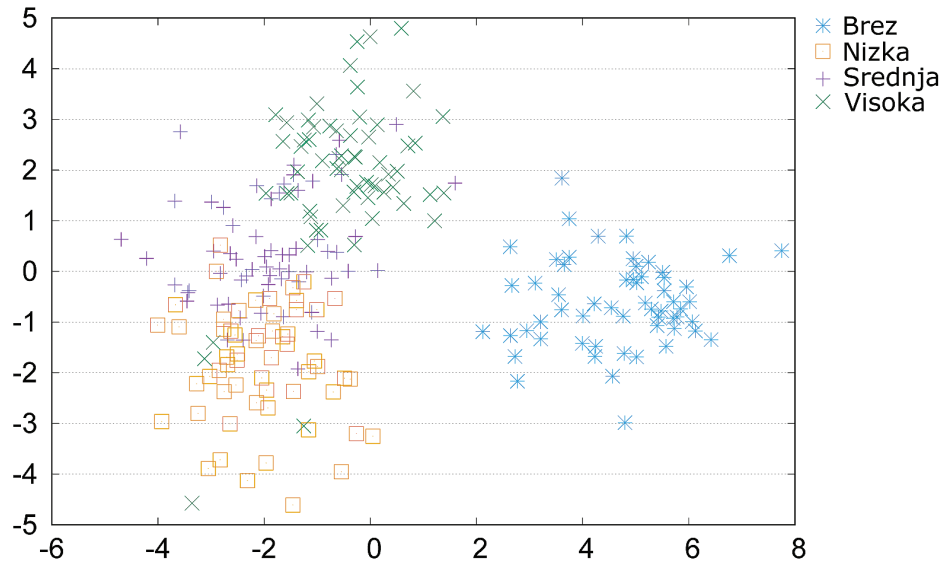
3.3 Napoved kognitivne obremenjenosti

Poglavitni cilj našega dela je preveriti povezavo med kognitivno obremenjenostjo in vedenjskimi vzorci uporabnika. S pomočjo strojnega učenja poskušamo napovedati trenutno stopnjo kognitivne obremenjenosti na podlagi vedenjskih (senzorskih) podatkov, kjer nam podatki možganske aktivnosti služijo kot preverba težavnostnih stopenj eksperimentalnih nalog.

Obe vrsti podatkov razdelimo na sekundo dolge časovne intervale in na podlagi posameznih intervalov izračunamo značilke. Tako za senzorske kot EEG podatke uporabimo hitro Fourierjevo transformacijo s Hannovim oknom na podlagi katere pridobimo moč signala med 0 in 100 Hz (IoT) ali 0 in 125 Hz (EEG). Pri senzorskih podatkih dodatno izračunamo značilke v časovni domeni. Podatki pridobljeni s pomočjo EEG so zaradi narave uporabljenih elektrod občutljivi na šum. Analiza spektrograma pokaže izrazit vrh pri 50 Hz in podobnega, vendar manjšega pri 100 Hz, ki nastaneta zaradi prisotnosti naprav, priključenih v električno omrežje. Za odstranitev tega šuma uporabimo zarezni (ang. notch) filter. Celoten seznam značilk je predstavljen v Tabeli 1.

Ustvarjena podatkovna zbirka skupaj vsebuje 2141 primerov. Vsak primer je označen z razredom naše ciljne spremenljivke, ki jo predstavlja trenutna stopnja kognitivne obremenjenosti na podlagi težavnosti naloge (odsotna, nizka, srednja, visoka). Zbirko razdelimo na posamezne uporabnike ter učno, validacijsko in testno množico (70 %, 15 %, 15 %). V zadnjem koraku, na podlagi učne množice, ločeno naučimo naključna gozdova s senzorskimi in EEG podatki (oba napovedujeta stopnjo kognitivne obremenjenosti). Naključni gozd v obeh primerih vsebuje

¹ <https://www.mentalab.com/>



Slika 1: Primer dvodimenzionalne LDA projekcije EEG podatkov enega uporabnika. Različne barve (oblike) predstavljajo različne stopnje kognitivne obremenitve. Prikažejo se jasne gruče podatkov z enako stopnjo kognitivne obremenjenosti. Ena točka predstavlja sekundo meritev.

1000 dreves, informacijski pribitek kot oceno kvalitete delitve drevesa in nima omejene največje globine drevesa. Izbrani parametri so določeni s pomočjo naključnega ter mrežnega iskanja in uporabo validacijske množice.

4 REZULTATI

V tem poglavju predstavimo analizo sposobnosti gručenja pridobljene podatkovne zbirke in rezultate, pridobljene na podlagi napovedovanja kognitivne obremenjenosti s pomočjo EEG čelade ter nato še s pomočjo senzorskih podatkov.

4.1 Analiza gručenja podatkov

Zmožnost gručenja podatkov v razrede različne kognitivne obremenjenosti nam poda prvi jasen znak ali lahko na podlagi pridobljenih EEG podatkov napovemo trenutno stopnjo kognitivne obremenjenosti. Podatke gručimo s pomočjo metode linearne diskriminantne analize (LDA). Primer gručenja po-

datkov za izbranega uporabnika je prikazan na Sliki 1. Opazimo lahko, da nastanejo jasne gruče primerov enake stopnje kognitivne obremenjenosti. Poleg tega lahko opazimo tudi jasno ločnico med primeri, ko uporabnik miruje ali opravlja nalogo. Podobni zaključki veljajo za vse uporabnike.

4.2 Napoved kognitivne obremenjenosti

Naslednji korak je napoved kognitivne obremenjenosti na podlagi podatkov zajetih z EEG čelade. Napovedujemo med štirimi različnimi stopnjami obremenitve. V tabeli napačnih klasifikacij, predstavljeni v Tabeli 2, opazimo močno diagonalo, saj je večina primerov razvrščenih v pravilni razred. Klasifikacijska točnost je 89,79 %, kar pomeni da zasnovane eksperimentalne naloge dobro vzpodbudijo različne stopnje kognitivne obremenjenosti. Opazimo tudi, da mnogo primerov, ki so klasificirani napačno, pade v sosednje razrede, kar prikaže zveznost kognitivne obremenitve.

Tabela 2: Tabela napačnih klasifikacij napovedi stopnje kognitivne obremenjenosti za model naključnega gozda na podlagi EEG podatkov.

Napovedan razred	Resnični razred			
	Odsotna	Nizka	Srednja	Visoka
Visoka	0	0	4	89
Srednja	0	11	72	0
Nizka	0	54	9	0
Odsotna	73	0	0	9

4.3 Analiza vedenjskih vzorcev

Na podlagi potrditve različnih stopenj kognitivne obremenitve s podatki možganske aktivnosti, opravimo enako napoved tudi s pomočjo senzorskih podatkov. Dosežena klasifikacijska točnost je v tem primeru 76 %. Vrednost je nižja kot pri EEG podatkih, vendar še vedno precej nad večinskim klasifikatorjem, ki doseže točnost 30,76 %.

Možen vzrok za razliko v točnosti med EEG in senzorskimi podatki leži v naravi zbranih podatkov. Vedenjski vzorci se spreminjajo precej bolj nenadoma, saj v trenutku ko oseba preneha pisati, inercialne merilne naprave ne zaznavajo več ničesar. Na drugi strani se izkaže, da imajo podatki o možganski aktivnosti večjo vztrajnost, saj krajši postanki med mislimi ne sprožijo nenadne spremembe kognitivne obremenjenosti. Kljub temu lahko potrdimo prisotnost vpliva kognitivne obremenjenosti na vedenjske vzorce posameznika, saj lahko v več kot treh četrtinah primerov pravilno napovemo stopnjo kognitivne obremenjenosti na podlagi senzorskih (vedenjskih) podatkov.

5 ZAKLJUČEK

V tem delu smo raziskali vpliv kognitivne obremenjenosti na vedenjske vzorce uporabnika, zajete z zbirko senzorjev interneta stvari. S pomočjo algoritma naključnega gozda smo napovedovali štiri različne razrede kognitivne obremenjenosti in dosegli klasifikacijsko točnost 76 %. Le-ta predstavlja znatno izboljšanje v primerjavi z večinskim klasifikatorjem (30,76 %). S tem smo pokazali, da kognitivna obremenjenost vsaj delno vpliva na vedenjske vzorce uporabnikov, saj je moč razlikovati med težavnostnimi stopnjami nalog zgolj na podlagi vedenjskih podatkov, zajetih s senzorjev. Da zasnovane naloge uporabnika dejansko potisnejo v stanja različne kognitivne obremenjenosti, smo enake napovedi opravili še s pomočjo podatkov možganske aktivnosti z EEG čelade. Tukaj je klasifikacijska točnost dosegla 89,79 %, kar potrjuje da so različne težavnostne stopnje nalog vzpodbudile različne stopnje kognitivne obremenjenosti pri uporabnikih.

Največja omejitev našega dela je velikost podatkovne zbirke. Zbirko nameravamo v prihodnosti razširiti z vključitvijo vsaj 100 prostovoljcev in dodatnimi senzorji, da bi lahko bolje razumeli vpliv kognitivne obremenjenosti na različne vidike obnašanja uporabnikov. Razširjeno podatkovno zbirko bomo

uporabili za izgradnjo sistema overjanja uporabnikov, ki se je zmožen prilagajati različnim stopnjam kognitivne obremenjenosti. Kot posledica majhne podatkovne zbirke se v tem delu omejimo na algoritem naključnega gozda. Večja bera podatkov nam bo omogočila uporabo bolj naprednih tehnik, vključno z algoritmi globokega učenja. To nam bo pomagalo pri doseganju višje klasifikacijske točnosti in avtomatski ekstrakciji značilk.

Prav tako moramo biti pozorni na šum, ki lahko izhaja bodisi iz merilnih naprav bodisi iz osredotočenosti udeleženca poskusov. Zaradi šibkosti signala so za šum dovzetne predvsem EEG čelade, saj lahko šum preglasi signal, ki ga želimo zajeti. To sploh velja v laboratorijskem okolju z veliko različne elektronske opreme. Pri udeležencih največji izziv predstavlja motiviranost za opravljanje nalog. Neosredotočenost na nalogo v našem eksperimentu nam, namesto veljavnih podatkov z različnimi stopnjami kognitivne obremenitve, lahko prinese podatke, kjer so udeleženci zgolj nizko kognitivno obremenjeni.

Pri nadaljnjem delu nam podatek o stopnji kognitivne obremenjenosti na podlagi podatkov s čelade EEG predstavlja referenčno vrednost in potrditev, da različne težavnostne stopnje nalog res različno vplivajo na stopnjo kognitivne obremenjenosti posameznika. Na podlagi tega smo lahko tudi prepričani, da so bili tudi senzorski podatki in posledično vedenjski vzorci uporabnikov, podvrženi različnim stopnjam kognitivne obremenjenosti. S pomočjo tako označenih podatkov lahko razvijemo sistem overjanja uporabnikov, ki je sposoben identifikacije ne glede na trenutno stopnjo kognitivne obremenjenosti. To bomo storili v prihodnosti na podlagi razširjene podatkovne zbirke in s pomočjo tehnike prilagajanja domeni [3] ter nevronske mreže z osnovno arhitekturo transformerjev [21].

6 LITERATURA

- [1] Shaibal Barua, Mobyen Uddin Ahmed, and Shahina Begum. Towards intelligent data analytics: A case study in driver cognitive load classification. *Brain Sciences*, 10(8):1–19, 2020.
- [2] Luis Cabañero, Ramón Hervás, Iván González, Jesús Fontecha, Tania Mondéjar, and José Bravo. Analysis of cognitive load using eeg when interacting with mobile devices. *Proceedings*, 31(1), 2019.
- [3] Youngjae Chang, Akhil Mathur, Anton Isopoussu, Junehwa Song, and Fahim Kawsar. A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–30, 2020.

- [4] Sarah J Davis, Steven J Schapiro, Susan P Lambeth, Lara A Wood, and Andrew Whiten. Behavioral conservatism is linked to complexity of behavior in chimpanzees (pan troglodytes): Implications for cognition and cumulative culture. *Journal of Comparative Psychology*, 133(1):20, 2019.
- [5] Cary Deck and Salar Jahedi. The effect of cognitive load on economic decision making: A survey and new experiments. *European Economic Review*, 78:97–119, 2015.
- [6] Cary Deck, Salar Jahedi, and Roman Sheremeta. The effects of different cognitive manipulations on decision making. *Economic Science Institute*, Working Paper, 2017.
- [7] Chao Feng, J I E Xiong, Liqiong Chang, Fuwei Wang, J U Wang, and Dingyi Fang. RF-Identity : Non-Intrusive Person Identification Based on Commodity RFID Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–23, 8 2021.
- [8] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, 2013.
- [9] Nir Friedman, Tomer Fekete, Kobi Gal, and Oren Shriki. EEG-based prediction of cognitive load in intelligence tests. *Frontiers in Human Neuroscience*, 13(June), 2019.
- [10] Paul Frosina, Michael Logue, Andy Book, Tylor Huizinga, Steven Amos, and Scarlet. Stark. The effect of cognitive load on nonverbal behavior in the cognitive interview for suspects. *Personality and Individual Differences*, 130(February):51–58, 2018.
- [11] Eija Haapalainen, Seungjun Kim, Jodi F. Forlizzi, and Anind K. Dey. Psycho-physiological measures for assessing cognitive load. In *UbiComp'10 - Proceedings of the 2010 ACM Conference on Ubiquitous Computing*, pages 301–310, 2010.
- [12] Magdiel Jiménez-Guarneros and Pilar Gómez-Gil. Custom domain adaptation: A new method for cross-subject, eeg-based cognitive load recognition. *IEEE Signal Processing Letters*, 27:750–754, 2020.
- [13] Andraž Krašovec, Daniel Pellarini, Dimitrios Geneiatakis, Gianmarco Baldini, and Veljko Pejović. Not Quite Yourself Today: Behaviour-based Continuous Authentication in IoT Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4), 2020.
- [14] Júlio Medeiros, Ricardo Couceiro, Gonçalo Duarte, João Durães, João Castelhana, Catarina Duarte, Miguel Castelo-Branco, Henrique Madeira, Paulo de Carvalho, and César Teixeira. Can eeg be adopted as a neuroscience reference for assessing software programmers' cognitive load? *Sensors*, 21(7), 2021.
- [15] Laura Mieth, Raoul Bell, and Axel Buchner. Cognitive load does not affect the behavioral and cognitive foundations of social cooperation. *Frontiers in Psychology*, 7, 2016.
- [16] Robert Morris and Ken Thompson. Password Security: A Case History. *Communications of the ACM*, 22(11):594–597, 1979.
- [17] Talha Ongun, Alina Oprea, Cristina Nita-Rotaru, Mihai Christodorescu, and Negin Salajegheh. The house that knows you: User authentication based on iot data. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 2255–2257, 2018.
- [18] Fred G. W. C. Paas, Jeroen J. G. van Merriënboer, and Jos J. Adam. Measurement of cognitive load in instructional research. *Perceptual and Motor Skills*, 79(1):419–430, 1994. PMID: 7808878.
- [19] Veljko Pejović, Tilen Matkovič, and Mojca Ciglarič. Wireless Ranging for Contactless Cognitive Load Inference in Ubiquitous Computing. *International Journal of Human-Computer Interaction*, 00(00):1–25, 2021.
- [20] Ioannis Stylios, Spyros Kokolakis, Olga Thanou, and Sotirios Chatzis. Behavioral biometrics & continuous user authentication on mobile devices: A survey. *Information Fusion*, 66(July 2020):76–99, 2021.
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5999–6009, 2017.
- [22] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. Human tracking and identification through a millimeter wave radar. *Ad Hoc Networks*, 116(December 2020):102475, 2021.

Andraž Krašovec je doktorski študent Fakultete za računalništvo in informatiko Univerze v Ljubljani. Trenutno je zaposlen kot raziskovalec na Skupnem raziskovalnem središču Evropske komisije. Njegovo delo se osredotoča na področje sistemov overjanja uporabnikov na podlagi vedenjskih vzorcev, zajetih s pomočjo tehnologij interneta stvari.

Veljko Pejović je doktoriral iz računalništva na Univerzi v Kaliforniji, Santa Barbara, ZDA. Po doktoratu je kot raziskovalec delal na Univerzi v Birminghamu, ZK. Od leta 2015 je zaposlen kot docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani (UL FRI), kjer vodi raziskave na temo mobilnega računanja, energetsko-účinkovitega računanja, računalniške varnosti, ter interakcije človek-računalnik. Nagrade, ki jih je dobil vključujejo Orange D4D challenge nagrado za modeliranje poteka epidemij s pomočjo podatkov iz mobilnih omrežij, ter Best paper nomination nagrado na konferenci ACM UbiComp. Več o njegovih raziskavah na <http://riss.fri.uni-lj.si/Veljko/>

■ Semantic reusable web components: A use case in e-government interoperability

Slavko Žitnik¹, Karmen Kern Pipan², Miha Jesenko², Dejan Lavbič¹

¹ Univerza v Ljubljani, Večna pot 113, SI-1000 Ljubljana

² Ministrstvo Republike Slovenije za javno upravo, Direktorat za informatiko, Tržaška cesta 21, SI-1000 Ljubljana
slavko.zitnik@fri.uni-lj.si, karmen.kern-pipan@gov.si, miha.jesenko@gov.si, dejan.lavbic@fri.uni-lj.si

Abstract

Advancements in technology and software engineering strive to efficiently build robust technologies and deliver them as fast as possible. It has been shown that both could be achieved by reusing existing implementations, libraries, components, and even frameworks. In the field of public administration, which needs to follow and implement national and EU regulations, it is essential to not only have compatible semantic data structures and processes but also to enable cross-sector and cross-border integrations. In this paper, we present the Digital Europe for All project, more specifically, a use case of a reusable prototype – a web component based on semantic data representation. We show that the component can be easily integrated into an arbitrary website and supports arbitrary evidence types, represented using the project's ontologies. Reusing such components enables faster creation of interoperable digital public services and opens new opportunities, better mobility, faster processes, and reduced implementation costs.

Keywords: Reusable components, building blocks, digital public services, reusability, software engineering, semantics

Ponovno uporabljive semantične spletne komponente: Primer uporabe za interoperabilnost digitalnih javnih storitev

Izvleček

Napredek v tehnologiji in programskem inženirstvu si prizadeva graditi učinkovite in robustne tehnike, ki bi se jih dostavljalo čimprej. Dokazano je bilo, da je to možno doseči s ponovno uporabo obstoječih implementacij, knjižnic, komponent ali celo ogrodij. Na področju javne uprave, ki mora slediti in implementirati nacionalne in evropske predpise, je poleg združljivosti na podatkovnem nivoju in procesov pomembno zagotoviti čezsektorsko in čezmejno integracijo. V tem prispevku predstavljamo projekt "Digital Europe for All," bolj natančno primer uporabe ponovno uporabljive spletne komponente, ki temelji na semantični predstavitvi podatkov. Prikažemo, da je komponento možno enostavno vgraditi v poljubno spletišče in podpreti poljubne tipe dokazil, ki so predstavljena s pomočjo projektne ontologije. Ponovna uporabljivost takšnih komponent bo hitreje zagotovila interoperabilne digitalne javne storitve in bo odprla nove priložnosti, boljšo mobilnost, hitrejše procese ter bo zmanjšala stroške implementacij.

Ključne besede: ponovno uporabljive komponente, gradniki, digitalne javne storitve, ponovna uporabljivost, programsko inženirstvo, semantika.

1 INTRODUCTION

Nowadays, we use many software systems based on large frameworks used by developers to build unifying architectures and easy-to-use graphical interfaces for the end users. Users leverage such software systems to support their business processes, and du-

ring that, they work on similar activities – e.g., entering data for multiple employees, showing diverse tables or updating data. *Building blocks* that enable business processes, for example, to enter the address of an employee or a company, should behave similarly. For software developers to achieve that, they need

to define common criteria for their software system to unify them and the users. *Building blocks* are so-called components, defined as follows: “An individual software component is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data).¹”

Software is rarely built from scratch, but to a great extent, some parts of code can be reused, copied and adapted to fit new requirements. For that reason, software process models were developed to provide guidance and create high-quality software at predictable costs. Most of the new software is thus based on the systematic use of well-defined components designed for various projects. Software engineering researchers defined the need for components already in the 1960s (McIlroy et al., 1968), while the subfield Component-based software engineering (CBSE) more widely developed and provided architectural solutions in the 1990s (Sametinger, 1997). During that time, prominent commercial computing infrastructure vendors followed mentioned ideas and presented their solutions for their programming frameworks, such as ASP.NET or Java (Kozaczynski and Booch, 1998).

Some researchers argue that focusing on independent components for modern software engineering is a mistake. Still, the focus should instead be on frameworks that even enable the existence of components (Wallace, 2010). The recent rise of Web programming frameworks showed precisely the opposite. These frameworks define components transpiled (i.e., translation of a source code to another language) into JavaScript, can be used completely independent of specifically used Web technologies and allow for custom parameterized user interface styling (Gackenheimer and Paul, 2015).

Large data economy systems that we can find in the public sector deal with enormous variety and amounts of data. This data is useless if it cannot be easily and correctly understood, accessed or combined. As software components can achieve interoperability and reuse on a technical level, they cannot solve problems on a semantic level. In that sense, governments pushed the development of open data initiatives where all public data would be described in a machine-readable format. Semantic Web technologies describe the data, provide unifying schemas,

and overcome problems of matching data fields only by short textual descriptions (Narducci et al., 2013).

The technologies mentioned above are the main building blocks of future interoperable systems. For the needs of globalization and cooperation of public administrations between EU member states, it is also crucial to enable semantic cross-sector and cross-border data interchange. Of course, privacy, security, storage, and other aspects of large software ecosystems need to be provided, but they are not the focus of this paper (GovStack, 2021).

This paper presents a use case for building a Web-based and semantically enabled reusable prototype - A web component for cross-border public administration interoperability. The rest of the paper is structured as follows. We first review technologies that have already tried to standardize and interconnect different public administrations in the EU. Then we continue with the brief presentation of the Digital Europe for All project

This article delves into specific use-case dealing with semantic and reusable prototype components as part of the field in the project group for Semantic interoperability solutions, where the authors of this article have been actively involved. We evaluate its adaptability and reusability and provide our view for further development.

2 RELATED TECHNOLOGIES SUPPORTING DIGITAL PUBLIC SERVICES

The previous section briefly overviewed general ideas for building reusable components in software engineering. A well-defined market segmented by Gartner already exists, and it is predicted that by 2024, 65 % of application development will be low code. In addition, recently, low-code and no-code platforms are becoming popular, leveraging reusable components. In this section, we focus on the state of provisioning the same set of functionalities to public administration.

Ideas for building reusable components in the public administration domain followed the development of CBSE in the 1990s (Castano et al., 1995). At that time, data was more tightly coupled with application logic, and therefore first approaches focused on identifying similar subschemas in Entity-Relational diagrams. Based on those, the development of common components was proposed. Also, most

¹ https://en.wikipedia.org/wiki/Component-based_software_engineering (Accessed: July 29, 2022).

of the work later focused on model-driven development for digital public services. There have been a lot of different models proposed by the UK, the EU and the USA to provide interoperability on various levels (Peristeras et al., 2008; Peristeras et al., 2009).

The United Nations member states defined 17 sustainable development goals representing the 2030 Agenda to connect all countries in a global partnership, improve health and education, reduce inequality, and spur economic growth. For the nations to even enable actions, their systems supporting digital public services must be interlinked. Apart from country-specific initiatives, multi-stakeholder alliances have also been formed that try to provide open-source software, models, content, software architectures and standards – e.g., *Digital Public Goods Alliance*, *GovStack* or *eGovernments Foundation*.

One of the key players in enabling and supporting cross-border and cross-sector interoperability between countries is the EU. It defines standards, connects multiple countries or stakeholders, and funds programmes and projects. Under the European Commission (EC) umbrella, several events are organized (e.g., SEMIC²) to promote and share new knowledge. EC also leads some projects (e.g., eSENS³, TOOP⁴) to address common topics or provide general frameworks that multiple stakeholders in member states can reuse (e.g., the DE4A project⁵, SDG⁶). In the last years, CEF Digital⁷, ISA, ISA^{2,8} and Interoperable Europe⁹ have delivered many valuable building blocks and solutions for public administrations to build interoperability at all levels, following European Interoperability Framework¹⁰.

In this regard, DE4A aims to establish a culture of co-creation, transparency, accountability and trustworthiness. Citizen and business-oriented pilots shall highlight chosen aspects of the technology ecosystem available for the Single Digital Gateway Regulation (SDGR¹¹) implementation at the European and Member State level, prove their technical viability and gauge the performance and degree to

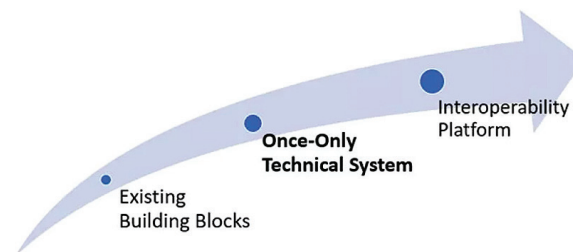


Figure 1: Stages towards a fully interoperable platform (source: <https://www.de4a.eu>, accessed July 29, 2022).

which non-functional requirements can be accommodated¹². We believe experiences, good practices, and information gained in the DE4A project will be of utmost importance for the further development of cross-border digital public services in member states and Slovenia.

3 DIGITAL PUBLIC SERVICES AND CROSS-BORDER INTEROPERABILITY

This section presents the details of the Digital Europe for All (DE4A) project, specifically related to the component we will more technically describe in the second part of the section. The DE4A project shows a possible path to full interoperability between different cross-border digital public services. The DE4A project started with the main building blocks supporting the SDGR's Once-Only Technical System (OOTS) and offers a way to interconnect cross-border public services. Figure 1 shows the stages of building such a platform.

3.1 The DE4A project

The Digital Europe For All (DE4A) project builds on prior work toward EU Digital Single Market to realize the Once-Only Principle. DE4A effectively puts forward a new Member State-driven large-scale pilot aimed at compliance with Single Digital Gateway and aligned with EU eGovernment Action Plan 2016-2020¹³, Tallinn Declaration and EIF Implementation Strategy. Its overarching goal is to reinforce trust in

² <https://www.de4a.eu> (Accessed: July 13, 2022)

³ https://single-market-economy.ec.europa.eu/single-market/single-digital-gateway_en (Accessed: October 12, 2022).

⁴ <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Digital+Home> (Accessed: July 14, 2022).

⁵ https://ec.europa.eu/isa2/home_en/ (Accessed: July 14, 2022).

⁶ <https://joinup.ec.europa.eu/collection/interoperable-europe/interoperable-europe> (Accessed: July 26, 2022).

⁷ https://ec.europa.eu/isa2/eif_en/ (Accessed: July 14, 2022).

⁸ https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.295.01.0001.01.ENG (Accessed: October 11, 2022).

⁹ https://ec.europa.eu/archives/isa/library/documents/eu-egovernment-action-plan-2016-2020_en.pdf (Accessed: July 29, 2022).

¹⁰ <https://eur-lex.europa.eu/legal-content/SL/TXT/?uri=CELEX%3A32016R0679> (Accessed: July 26, 2022).

¹¹ https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.257.01.0073.01.ENG (Accessed: July 26, 2022).

¹² <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32006L0123> (Accessed: July 26, 2022).

¹³ <https://www.de4a.eu/about-project>, (Accessed: July 21, 2022).

public institutions and to unleash multiple measurable positive impacts in terms of efficiency gains and reduction of current administrative burden and costs, rooted in a Toolkit for extended semantic interoperability and in the secure, privacy-preserving and trustworthy realization of fundamental once-only, relevant-only and digital by default principles, through state-of-the-art, usable and high-quality fully online procedures accessible through the Single Digital Gateway defined in SDGR. The DE4A takes reality as its starting ground - the needs and the capacities of the Member States (MS), ensures compliance with all other relevant regulations (GDPR¹⁴, eIDAS¹⁵, Services Directive¹⁶, among others) as well as EU and MS strategic eGovernment and Single Market goals and aims to transform governance towards open collaboration and innovation, adaptable to depart from the variety of current environments¹⁷. It will allow for full regulatory compliance, establishing a culture of co-creation, transparency, accountability and trustworthiness. For a completely working digital single market, effectively enabling the cross-border exercise by citizens and businesses of their Single Market rights, EU MS must address several challenges in delivering better services. DE4A is a three-year pilot-led project started on January 1 2020, in support of the SDG across Europe ((EU) 2018/1724) and helping to make the single digital market a reality. DE4A includes 22 partners, including eight Member States (Austria, Luxembourg, Netherlands, Portugal, Romania, Slovenia, Spain, and Sweden). The project establishes a culture of co-creation, transparency, accountability and trustworthiness. Its goal is to facilitate migration towards European Digital Public Services co-delivered across borders, across sectors and with different participants. The project has received funding from the European Union's Horizon 2020 research and innovation program under GA No. 870 635¹⁸. ATOS leads the DE4A project, and it consists of several different work packages (project groups) covering different areas as follows: Inventory of current eGovernment landscape; Architecture vision and framework; Semantic interoperability solutions; Cross-border pilots for citizens, business and evalu-

ation; Common component design & development; Sustainable impact and new governance models; Legal and ethical compliance and consensus building; Stakeholder dialogue, dissemination and communication; Project coordination and management and Ethics requirements.

The project is validating a comprehensive and scalable approach to develop and demonstrate the potential and benefits of securely and digitally sharing relevant-only and once-only data. DE4A aims to establish a culture of co-creation, transparency, accountability and trustworthiness. Citizen and business-oriented pilots shall highlight chosen aspects of the technological ecosystem available for the SDGR implementation at European and MS levels, prove their technical viability and gauge the performance and degree to which non-functional requirements can be accommodated. The pilots comprise functional use cases requiring interaction with real users – citizens, students, businesspersons and public servants. The three pilots are as follows:

- **Studying Abroad Pilot** will enable paperless procedures for students' mobility and simplify cross-border interactions for students engaging with processes, systems and platforms used by higher education establishments. It will allow digital (a) applications for higher education, (b) applications for study grants, (c) and diplomas/certificates/studies/ professional recognition.
- **Doing Business Abroad Pilot** will lower barriers for companies to start a business or conduct business in another EU MS. It will enable businesses to (a) retrieve and keep up-to-date company data from authentic sources, (b) start a business and (c) digital annual reporting.
- **Moving Abroad Pilot** will enable EU citizens' mobility across the EU and support them in moving from one MS to another by creating a digitized process for exchanging information required to facilitate a change of residency and other online cross-border procedures relying on civil status certificates.¹⁹ The processes will support (a) registering change of address, (b) civil status certificates, and (c) pension claims and information.

¹⁴ https://www.de4a.eu/_files/ugd/2844e6_1e295e0aa15c4afa8f7f1baa52001189.pdf, (Accessed: July 21, 2022).

¹⁵ https://www.de4a.eu/_files/ugd/2844e6_1e295e0aa15c4afa8f7f1baa52001189.pdf, (Accessed: July 21, 2022).

¹⁶ https://wiki.de4a.eu/index.php/Birth_Canonical_Evidence (Accessed July 26 2022).

¹⁷ <https://github.com/de4a-wp3/MOR> (Accessed: July 15, 2022).

¹⁸ https://www.de4a.eu/_files/ugd/2844e6_1e295e0aa15c4afa8f7f1baa52001189.pdf, (Accessed: July 21, 2022).

¹⁹ https://www.de4a.eu/_files/ugd/2844e6_1e295e0aa15c4afa8f7f1baa52001189.pdf, (Accessed: July 21, 2022).

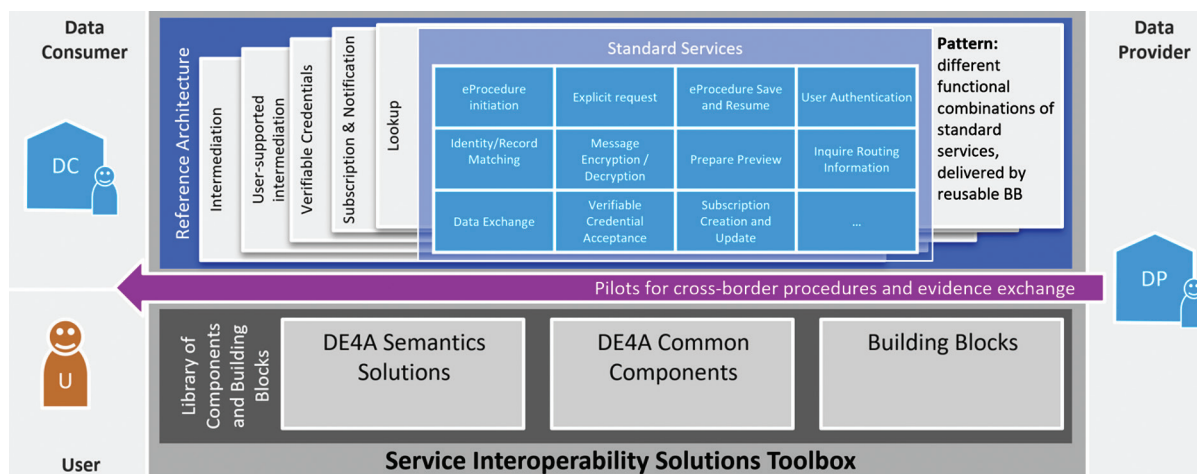


Figure 2: The DE4A project Service Interoperability Toolbox (source: <https://wiki.de4a.eu>, accessed July 29, 2022).

The project’s main deliverables will comprehensively document and validate technologies that must be implemented for the cross-border integration of public services. Figure 2 shows the central and long-term deliverable of the proposed architecture framework. Some of the final deliverables will define all aspects, such as modelling requirements, processes, semantic solutions with models, communication messages and patterns and full framework implementation. The results will be demonstrated within the production-level implementation of specific cross-border and cross-domain pilots described above.

The key business roles identified by the DE4A are **User**, **Data Consumer (DC)**, which consists of **Data Evaluator (DE)** and **Data Requestor (DR)**, **Data Provider (DP)**, which consists of **Data Transferor (DT)** and **Data Owner (DO)**, and **Registrar**. User means a natural or legal person that accesses the system (uses the service). The registrar is responsible for accepting and storing formal information according to rules of law or governance. Data Consumer is an organization/administration in demand of the data, while a Data Provider is a legal entity in charge of the data deployment. It is essential to understand their specialized roles:

- **Data Consumer (DC)**
 - **Data Evaluator (DE)**: organization authorized to receive and process data from a User via OOTS.
 - **Data Requestor (DR)**: a technical role that searches and requests data. If DR is a separate entity from DE, it carries the request under the mandate of the DE.
- **Data Provider (DP)**

- **Data Owner (DO)**: organization which governs information about the User. DO is responsible for authorization approval, data extraction (also from secondary registries) and audit control.

- **Data Transferor (DT)**: a technical role responsible for the actual data transmission. It runs the OOTS-compliant data service. DT and DO can be different entities – e.g. when a national or sectorial intermediary is acting as an evidence broker.

These business roles can exchange structured data (i.e., evidence) or related information following Once-Only Interaction Patterns. The DE4A Architecture Framework defines five reference interaction patterns:

1. Intermediation Pattern,
2. User-supported Intermediation Pattern,
3. Verifiable Credentials Pattern,
4. Lookup Pattern, and
5. Subscription and Notification Pattern.

As the reusable prototype component we describe in this paper uses only the User-supported Intermediation Pattern or Intermediation Pattern, we will present these only. Other patterns are primarily used for communication between systems or credentials-related needs. Figure 3 shows an illustration of the Intermediation Pattern. In this pattern, a user initiates a procedure through a public service request. The DC is missing the required evidence that can be automatically acquired from a known DP. After the evidence (one or more) is retrieved, they are shown to the user, who can decide to complete the procedure. In some cases, the preview part might also allow



Figure 3: Illustration of the Intermediation Pattern.

a user to upload signed evidence in his possession manually.

In Figure 4, we show an example of a User-supported Intermediation Pattern. In this pattern, a user initiates a procedure through a public service request. In case when a DC needs additional evidence, it notifies a user and redirects him to identify and authenticate himself with a selected DP. In the meantime, the DC has requested the missing evidence, shown to the user via a preview component. If the user approves the evidence, DP sends it to the DC. If DC is missing more evidence, this process will be repeated. When all evidences are acquired, the user can decide to submit the initiated procedure.

3.2 Reusable prototype component use case: Multilingual Ontology Repository (MOR)

In the scope of the DE4A project, we have designed and tested the Multilingual Ontology Repository (MOR) prototype component for possible further

use in semantic data exchange among different EU member states. The MOR component aims to register and provide information on concepts and terms involved in the evidence exchange and support the automatic generation of customizable user interfaces (i.e., for explicit request, preview, and additional parameters request) that contain complex terms in any EU official language. The component is developed as a prototype and is meant to be integrated into an arbitrary organization/administration portal that offers functionalities to users (e.g., evidence preview in step 5 of User-supported Intermediation Pattern communication).

The DE4A defined clear semantic and functional requirements for the component. Semantic requirements are highly related to MOR terms which are semantic elements (simple or complex) represented by a Uniform Resource Identifier (URI), syntax and multilingual meaning. Examples of complex types are canonical evidence types (e.g., birth certificate²⁰)

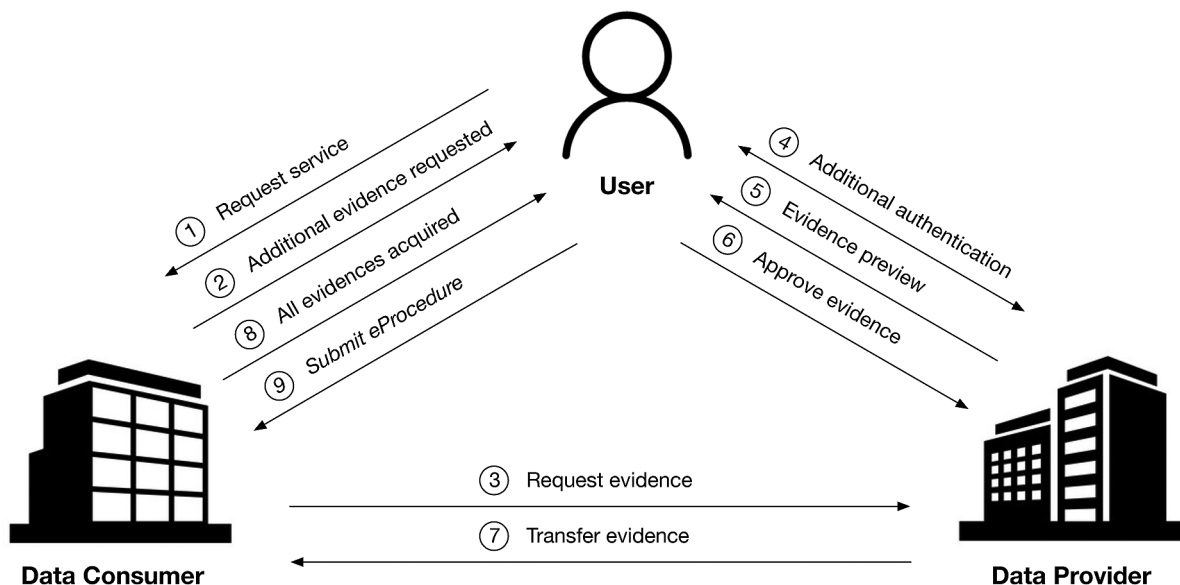


Figure 4: Illustration of the User-supported Intermediation Pattern (* in a case when additional evidence is requested).

that MOR needs to be able to process. Each term must be described using a label, description and an example in every EU official language. Suppose a description and an example are not provided for the target language. In that case, the MOR component needs to translate values automatically – these values must be shown to the user using the “non-verified” label. Complex terms are modelled as a tree hierarchy of terms, and a path within the hierarchy uniquely identifies them. If a complex term [X] is defined of type [Y], the [X] sub-terms have the same URI as the [Y] sub-terms except for the root element, so the root of [Y] term paths is replaced by the [X] URI. For example, the sub-term “Gender” of the complex term “*BirthEvidence/Child*” of type “*Person*” corresponds to the “*Person/Gender*” sub-term but with URI “*BirthEvidence/Child/Gender*.” MOR also contains code lists for enumeration types. All terms must be reusable. For example, the complex term “*BirthEvidence/Child*” of type “*Person*” may overload the multilingual meaning label and description for newborn persons instead of persons in general.

Apart from semantic and syntax support, MOR must enable the automatic generation of customizable user interfaces for any complex term in any EU official language. There are three cases where this functionality can be of help:

- The explicit request functionality should inform the user of the information to be requested as evidence, so the MOR can help to create a building block that generates such a user interface for any canonical evidence type and language.
- The preview functionality should show the user the evidence to be incorporated into the procedure, so the MOR can help to create a building block that generates such a user interface for any canonical evidence type and language; audits can also use this building block to help auditors to understand any canonical evidence in any language. This is particularly interesting when the preview space is located on the evidence provider side. The language on this side can differ from the procedure requiring cross-border evidence, and the user may not understand the provider’s language.
- The additional parameters functionality requires the user to request some fields through a form,

so the MOR can help create a building block that generates such a form for any additional parameters and language set. This is of particular interest when additional parameters have to be required on the evidence evaluator side because the evidence provider sets these parameters. In this case, the type of the terms is the key to generating the appropriate input field in the form (calendar, select list, text box, etc.)

3.3 Technical implementation

In this section, we overview the implementation of the prototype – MOR component, show options for its possible usage, and comment on its reusability and extensibility. The implementation is publicly available in a GitHub repository²¹.

MOR implementation has two aspects: (a) the storage of the MOR terms according to the specification, and (b) the standard Graphical User Interface (GUI) components to provide, in a multilingual way, a customizable user interface for the explicit request, evidence preview and requesting additional parameters for the record matching. Within the DE4A project, GUI for requesting additional parameters has not been considered, and some simplifications have been adopted to implement the MOR storage to reduce the technical complexity of its implementation. Since only the DE4A Moving Abroad pilot will use the component, MOR storage contains only canonical evidence types for Birth (*BirthEvidence*) and Marriage (*MarriageEvidence*).

3.3.1 Simplified MOR storage

MOR needs to store all MOR terms required for the three functionalities. As in the production environment, evidence types can be modified, or new types can be added; MOR should access these types via centralized services.

In our case, MOR storage is implemented as JSON files, one per available language, offered by a REST API. After the modelling, we generated files that represent all supported canonical evidence types (i.e., *BirthEvidence* and *MarriageEvidence*), complex terms used as types of other complex terms (e.g., *Person*, *Location*), and code lists (e.g., the EU Member States, administrative-territorial units, languages

²⁰ https://wiki.de4a.eu/index.php/Birth_Canonical_Evidence (Accessed July 26 2022).

²¹ <https://github.com/de4a-wp3/MOR> (Accessed: July 15, 2022).

codes, currencies, gender). Available languages are the ones related to the Member States that participate in the Moving Abroad pilot: French (fr), Portuguese (pt), Slovenian (sl), Spanish (es), and, as the common language, English (en).

Each MOR term is comprised of five elements:

1. **Term URI:** URI is the path corresponding to the XML element of the term in the XML schema of the complex. For instance, the root MOR term of the *BirthEvidence:1.0* canonical evidence type would be “*BirthEvidence*”, i.e., the root element of the corresponding XML schema. The rest of the sub-elements of such XML schema are specified as separate MOR terms, such as “*BirthEvidence/CertifiesBirth/Child*.”
2. **Origin:** Source of the type of term, either DE4A or other XML namespaces, such as the ones used by ISA². As the MOR component does not need to be aware of its origins, this field is not included in the MOR schema.
3. **Type:** Depending on the nature of the term, according to the defined schema:
 - XML base type with prefix “xsd:” (e.g., xsd:string) for a simple term.
 - “xsd:enumeration” for a complex term defined as a code list.
 - “xsd:token” for a simple term defined as an entry of a code list.
 - URI of another MOR term specified in a different schema (e.g., MOR term *LocationAddress* as

a type of the MOR term *BirthEvidence/CertifiesBirth/Child/PlaceOfBirth*).

- No type for root MOR terms. A complex term whose sub-elements are included in the same schema, so their URI starts with the URI of the complex term (e.g., *BirthEvidence*).
4. **Cardinality:** Represented as two Boolean values. The first value represents if the term is mandatory, and the second represents if the term can have multiple values. Root MOR terms have no cardinality.
 5. **Comments:** Optional text to describe the term specification.

3.3.2 Multilingual Ontology Repository Application Programming Interface (MOR API)

The MOR API is defined to be the same for future implementations of the MOR storage with an existing database system. The MOR API retrieves all MOR terms for a given language via one REST endpoint: “*/mor/{lang}*.”

The JSON object returned by the MOR API contains MOR term values (presented in the previous section) and language-specific type values. These values are needed as metadata to present the retrieved schema to the user. The “*{lang}*” JSON object consists of a label, a description, an example, and a verified flag.

Examples of MOR term objects when Spanish is the selected language:

```
"BirthEvidence": {
  "type": "",
  "cardinality": "",
  "comment": "",
  "es": {
    "label": "Prueba de nacimiento",
    "description": "Información sobre un nacimiento que prueba su registro.",
    "example": "",
    "verified": "true"
  }
},
"BirthEvidence/CertifiesBirth/Child/PlaceOfBirth": {
  "type": "Person/PlaceOfBirth",
  "cardinality": "10",
  "comment": "Country and place of birth",
  "es": {
    "label": "País y lugar de nacimiento",
    "description": "País y lugar donde nació un niño",
    "example": "Spain, Cercedilla",
    "verified": "true"
  }
}
```


Note that “BirthEvidence” is the root term in Birth Canonical Evidence Type, so values of “type” and “cardinality” are empty.

The JSON specification of the MOR API response is aimed at easing the finding of MOR terms when using a JavaScript JSON variable. Because of this reason, the MOR specification of the term with URI “BirthEvidence/CertifiesBirth/Child/PlaceOfBirth” can be easily obtained by the expression `javascript:JsonVar[URI]`.

3.3.3 Reusable and customizable MOR GUI

Explicit request and evidence preview are required functionalities by every DE and DO when using a

User-supported Intermediation Pattern. Multilanguage support is vital as a user from any member state can use services in any other member state.

As evidence types were thoroughly defined using semantics, schemas make it possible to define automatically customizable GUI. The look and feel of the component can be completely changed by providing a customized Cascading Style Sheet file (CSS). Each HTML DOM object within MOR is identified by its id; therefore, an integrator can completely redesign the component’s user interface. The component is implemented as an AngularJS component and can be included into any Web page

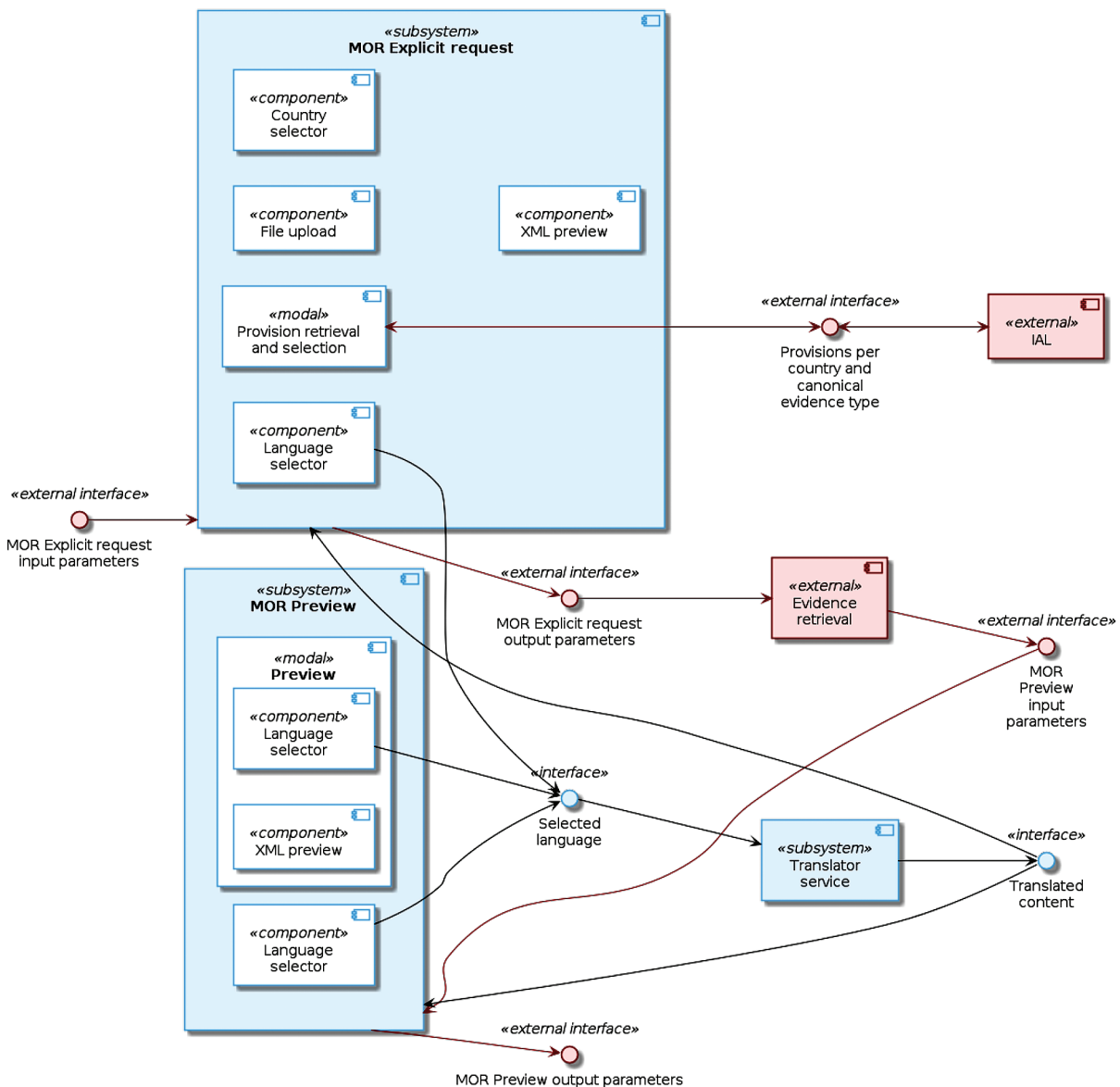


Figure 5: Multilingual Ontology Repository component diagram.

using HTML placeholder `<app-mor-er ...>/app-mor-er</app-mor-er>`.

In Figure 5, we show the MOR component diagram. The blue denotes MOR implementation, while the red represents the external system (i.e., a specific Web page) where the MOR component could be integrated. MOR component requires a definition of a few parameters to be placed within the target Web page. Results of the component are returned into a JavaScript variable within the embedded Web page (as parameterized during placement). Below, we describe the configuration and running of explicit request and evidence preview functionalities.

```
<app-mor-er
  defaultLang = "en"
  requesterCountryCode = "BE"
  canonicalEvidenceTypes = "BirthEvidence,MarriageCertificate"
  outputJSMapId = "outputJSMapIdMorEr"
></app-mor-er>
```

When the page with the embedded MOR component is open, the MOR API is called to retrieve the MOR JSON in the given language. Figure 6 shows a default MOR GUI for the explicit request functionality (with the default language set in Slovene). It allows for changing the default language and provisioning of the needed evidence. For each evidence it provides for:

- Review of the schema and data fields that constitute evidence. The review is available in multiple languages and provides a list of all labels and descriptions of the fields (see Figure 7).

Explicit request

A page of the DE portal opens the MOR GUI page for the explicit request. The parameters that DE needs to provide when embedding the component into a Web page are the following: (a) a default component language, (b) the DE country, (c) a list of Canonical Evidence Types identified by the MOR root terms, and (c) the name of a JavaScript Map variable to fill by the MOR component (the hosted Web page can then use this data). An example of embedding a MOR component for the explicit request functionality:

- Providing the evidence to the DE. The user can upload each evidence if he owns a relevant and verifiable digital document accepted by the DE (e.g., XML, PDF). The DE can automatically retrieve evidence if the user selects the country that takes the role of a DP. The evidence retrieval process follows the predefined exchange patterns (described in the previous section).

The selection made by the user per canonical evidence type is returned by filling in the JavaScript Map variable specified. The keys in the map variable

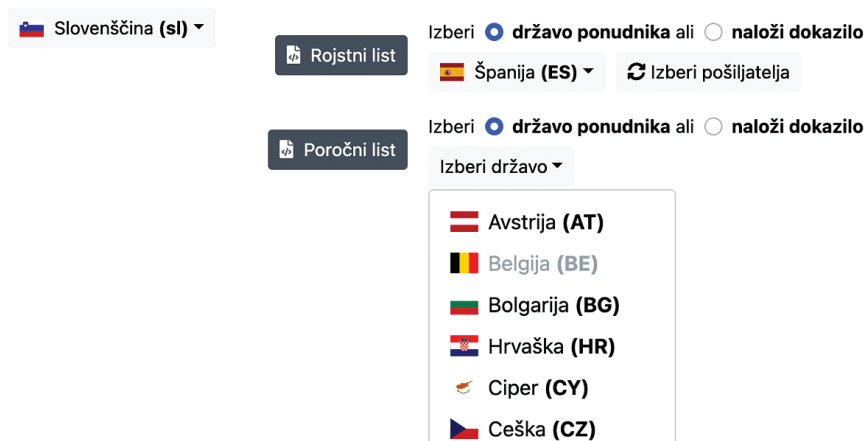


Figure 6: Default MOR component GUI for the explicit request functionality.

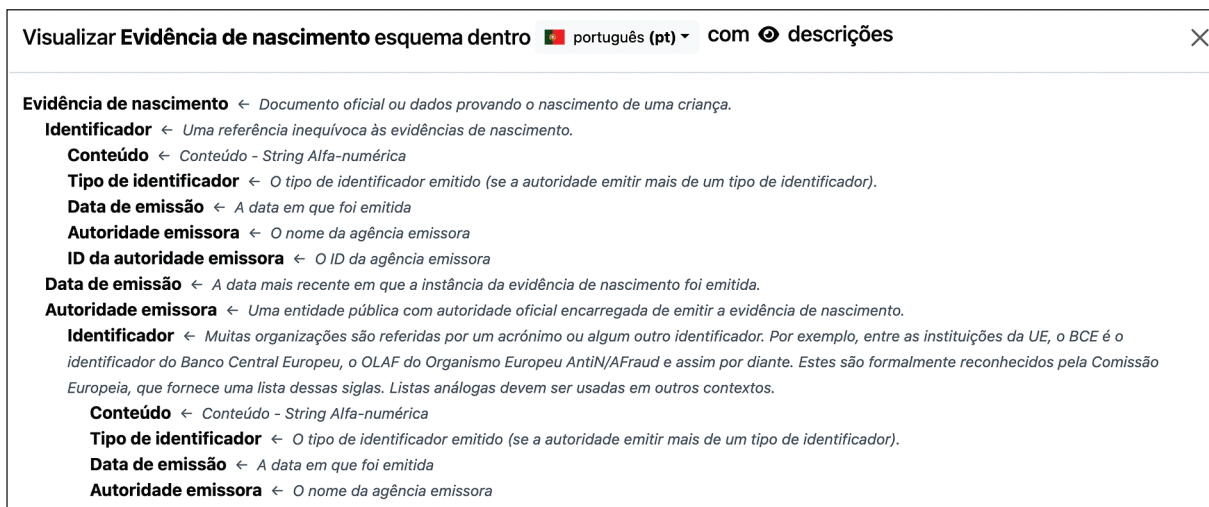


Figure 7: Selected evidence type preview within a default MOR GUI.

are the needed evidence type names. Each entry consists of one from two possible values depending on the user choice, that is, (a) the selected provider country or (b) the ASCII representation of the uploaded file, which can be either an ASCII file such as an XML file or a binary file such as a PDF.

Evidence Preview

After issuing the requested canonical evidence, the MOR component GUI for the evidence preview is opened by a Web page at the DE or the DO side, depending on the type of the exchange pattern (described in the previous section).

When the page with the embedded MOR component is open, the MOR API is called to retrieve the MOR JSON in the given language (if not already cached by the browser). Figure 8 shows a default MOR GUI for the evidence preview functionality. The parameters for embedding the preview part of the component include (a) the default component language, (b) a JSON array that contains JSON objects of specific evidence (key is the evidence type name and value is its representation), and (c) a JavaScript

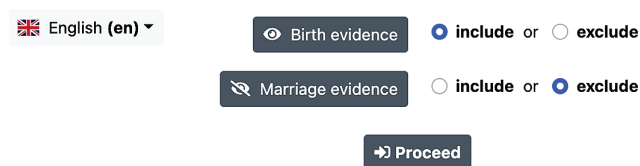


Figure 8: Default MOR component GUI for the evidence preview functionality.

Map variable to fill by the MOR component with the result of the evidence preview. For instance:

```
<app-mor-p
  defaultLang = "en"
  postActionValue = "[{...}, {...}]"
  outputJSMapId = "outputJSMapIdMorEr"
></app-mor-p>
```

The component allows for reviewing the retrieved evidence types from the DO. If the evidence is not provided in machine-readable XML format following semantic schemas, the evidence is available as a binary file (e.g., PDF, image). The preview of evidence is like the explicit request part (Figure 7) with the addition of actual values for the user's evidence. Users can exclude the evidence at this step, and within the process, the system will ask them again to provide the needed canonical evidence (i.e., explicit request part).

The selection made by the user per canonical evidence type is returned by filling the defined JavaScript Map variable. The key to each entry is an evidence type name, while the value is of Boolean type, depending on the user's choice to include the evidence within the service.

3.4 Reusability evaluation

The MOR component building blocks (i.e., explicit request and evidence preview) have the advantage of being reusable and generic for any complex term and language, so parties in the evidence exchange do not need to develop their equivalent components, and any

modification in MOR is automatically available through the MOR building blocks. For the best integration from the user experience perspective, CSSs are used. More specifically, we define the following aspects of adapting the component and its reusability:

- Configuration and integration: The component is packaged as a self-contained HTML DOM object with JavaScript functionality. It can be included on any modern Web page. The parameters for the component are set at its placement into an HTML DOM by providing values for its attributes. The results of the component are outputted to a selected JavaScript variable.
- Support of additional evidence types: The component can support an arbitrary evidence type and an arbitrary number. This is enabled using semantic schemas and MOR terms JSON provided by the DE4A services. If the services update JSON responses, the MOR component will automatically consider them and adapt the preview.
- Multilanguage support: Multiple languages support is enabled by MOR terms JSON files that are offered by the DE4A services. This JSON also includes translations for MOR component-related translations (e.g., names of buttons) which start with the root MOR term "GUI." For example, a

label of a proceed button in a selected language is available in the "GUI/proceed" field.

- Visual integration: One of the essential requirements was to support fully customizable visual integration of the component. There can be many DEs or DPs from different countries, and they already have their Web sites supporting digital public services. Therefore, the need is that the MOR component can be adapted for each one of those portals. In that sense, all the HTML DOM components in MOR have an associated id, based on which visual characteristics can be defined via the MOR CSS during the integration.

Furthermore, we would like to mention the Slovenian national "Once-Only" platform "Tray", which operates as a decentralized data exchange system supporting the implementation of the Government to Citizens (G2C), Government to Businesses (G2B) and Government to Government (G2G) e-procedures. The system can also obtain data from many sources through a single-entry point, exchanging a minimum data set. The central system, "The tray", takes care of optimal data load from all the data sources with the help of the control and forecasting system based on continuous machine learning. Based on this technology, a prototype for Slovenian evidence preview

REPUBLICA SLOVENIJA Ana Maček ID: 0707993506389

Pladenj Preview

Preview

The following information has been retrieved from the business register. Please review the provided information.
The information is formatted according to the European standards, therefore some information might look different from the country specific presentation.

HIGHER EDUCATION DIPLOMA			
TITLE	Masters in dance	DEGREE	Masters
COUNTRY	http://publications.europa.eu/resource/authority/country/SVN	INSTITUTION NAME	Academy for Dance
STUDY PROGRAMME	Dance, choreography	MAIN FIELD OF STUDY	http://data.europa.eu/snb/iscsd-1/0215
MODE OF STUDY	http://data.europa.eu/europass/learningScheduleType/fullTime	DURATION OF EDUCATION	P2Y
SCOPE	120	DATE OF ISSUE	2017-04-21

PLACE OF ISSUE

NAME	Ljubljana
------	-----------

HOLDER OF ACHIEVEMENT

NATIONAL ID	0707993506389	GIVEN NAMES	Ana
FAMILY NAMES	Maček	DATE OF BIRTH	1993-07-07

Reject data Accept data

Figure 9: Evidence preview example by the Slovenian implementation of a pilot supporting digital public services related to "Studying abroad" – Pladenj (Tray) Preview.

has been developed in the DE4A project as one of the possible solutions. Figure 9 shows an example of evidence preview by the Slovenian implementation of a pilot supporting digital public services related to “Studying abroad” (using Tray for data exchange). MOR component can be adapted to allow the same look and feel (in comparison to the default MOR evidence type preview setting, shown in Figure 7).

MOR component might therefore be used by countries that do not yet implement such functionalities. Within the DE4A project, these countries might be Spain, Portugal or Romania, which take part in the pilots.

4 CONCLUSIONS

In this paper, we presented the main characteristics of the DE4A project aiming to support the delivery of cross-border digital public services. We showed the evolution of the development of reusable prototype components in software engineering and how it evolved for the needs of public administration systems based on OOTS principles. We briefly reviewed progress in cross-border and cross-sector public administration interoperability. We focused on specifics of Digital Europe for all projects, especially those related to the use case of a reusable prototype component. In more detail, we presented a use case for building a semantic and Web-based reusable prototype component. The pilot component Multilingual Ontology Repository is a deliverable of the Digital Europe for all project. We evaluated and showed how the component offers adaptability to any Web site and is configurable to support future semantic models.

We believe that using such prototypes and building blocks could support the EU Member States’ public administrations’ interoperability and would open new opportunities to support mobility, fasten processes and be very cost-effective. Also, innovative technologies will be able to serve with transformative impact, for example, blockchain (for effective notarization, fostering accountability and transparency in the distributed transactional environment) and machine learning (over usage data to automate monitoring and improve effectiveness/quality of eProcedures).

5 ACKNOWLEDGEMENTS

The research and development reported in this paper have received funding from the European Union’s Horizon 2020 Research and Innovation Programme under grant agreement no. 87035 (DE4A: Digital Europe for All). We would like to thank the members of the DE4A WP3 group, especially the WP leader, Thashmee Karunaratne, and Ana Rosa Guzmán. We would also like to give credit to the authors of the DE4A deliverable D2.1, from where we used Figure 2 (ATOS is the leading partner of the DE4A project).

6 LITERATURE

- [1] Sametinger, J. (1997). Software engineering with reusable components. *Springer Science & Business Media*.
- [2] Kozaczynski, W., & Booch, G. (1998). Component-based software engineering. *IEEE Software*, 15(5), 34-36. doi:<https://doi.org/10.1109/MS.1998.714621>.
- [3] McIlroy, M. D., Buxton, J., Naur, P., & Randell, B. (1968, October). Mass-produced software components. In *Proceedings of the 1st international conference on software engineering*, Garmisch Pattenkirchen, Germany (pp. 88-98).
- [4] Wallace, B. (2010). A hole for every component, and every component in its hole. *Existential Programming*.
- [5] Gackenheim, C., & Paul, A. (2015). *Introduction to React* (Vol. 52). Apress.
- [6] Narducci, F., Palmonari, M., & Semeraro, G. (2013). Cross-language semantic matching for discovering links to e-gov services in the lod cloud. *KNOW@ LOD*, 992, 21-32.
- [7] GovStack (2021). Accelerating the digital transformation of government services. <https://www.govstack.global>. (Accessed: July 13, 2022).
- [8] Castano, S., Antonellis, V. D., & Pernici, B. (1995, August). Building reusable components in the public administration domain. In *Proceedings of the 1995 Symposium on Software reusability* (pp. 81-87).
- [9] Peristeras, V., Tarabanis, K., & Goudos, S. K. (2009). Model-driven eGovernment interoperability: A review of the state of the art. *Computer Standards & Interfaces*, 31(4), 613-628.
- [10] Peristeras, V., Loutas, N., & Tarabanis, K. (2008, March). Organizational engineering in public administration: the state of the art on eGovernment domain modeling. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 580-587).
- [11] Digital Europe for All, <https://www.de4a.eu/> (Accessed: July 21, 2022)

■

Slavko Žitnik is an assistant professor at the Faculty of Computer and Information Science, University of Ljubljana. His main research interests are semantic Web and natural language processing, specifically text mining, information retrieval and information extraction. He is in active collaboration with the industry on his research topics and has active collaborations with Harvard University, University of South Florida, and University of Belgrade.

■

Karmen Kern Pipan is the Head of Data Governance Department, Office for Development of Digital Solutions, Information Technology Directorate at the Ministry of Public Administration of Republic Slovenia. She is involved in data management, business intelligence, big data analytics, semantics and interoperability solutions to improve data-driven decision-making and user experience in public administration. She led an inter-ministerial task force to prepare the Public Administration Development Strategy 2015-2020.

■

Miha Jesenko has a legal-based background with Degree in Law, upgraded by Masters in Information Technology and Law. He works in Data Governance Department, Office for Development of Digital Solutions, Information Technology Directorate at the Ministry of Public Administration of Republic Slovenia. His main areas of work relate to data governance and interoperability, especially semantic interoperability through work on ontologies, controlled vocabularies, and reference data. An important part of his work is cooperation in efforts to introduce new technologies used by public administration.

■

Dejan Lavbič is an associate professor at the Faculty of Computer and Information Science, University of Ljubljana. His research areas include Information Quality, Semantic Web and Web3, distributed ledger technologies and self-sovereign identity. He collaborated in numerous EU and national-funded projects and has a long track of collaborations with the industry

Iz Islovarja

Islovar je spletni terminološki slovar informatike, ki ga že več kot 20 let ureja jezikovna sekcija Slovenskega društva INFORMATIKA. Slovar najdete na naslovu <http://www.islovar.org>. Tukaj objavljamo izbor izrazov, ki jih je strokovna skupina urednikov uredila s pregledom vsebine standarda ISO 27000.

celovitost -i ž (*angl. integrity*) lastnost, ki zagotavlja točnost in popolnost informacijskega sistema, storitve; sin. neokrnjenost, integriteta

CIA cíje krat. ž (*angl. confidentiality, integrity and availability, CIA*) osnovna načela informacijske varnosti: zaupnost, celovitost, razpoložljivost

informacijska várnost -e -i ž (*angl. information security*) ukrepi in postopki, ki zagotavljajo ohranjanje zaupnosti, celovitosti in razpoložljivosti informacij; prim. varnost informacijskega sistema, INFOSEC

istovétnost -i ž (*angl. authenticity*) verodostojnost podatkov o identiteti; sin. avtentičnost; prim. identiteta

neskládnost -i ž (*angl. nonconformity*) neizpolnjevanje postavljenih zahtev

ranljívnost -i ž (*angl. vulnerability*) lastnost sredstva, sistema, ki lahko omogoči uresničitev varnostne grožnje

razpoložljívnost -i ž (*angl. availability*) lastnost, ki zagotavlja dostopnost in uporabnost sredstva, sistema; prim. dostopnost

skládnost -i ž (*angl. compliance*) lastnost, ki zagotavlja izpolnjevanje postavljenih zahtev

zanesljívnost -i ž (*angl. reliability*) lastnost sredstva, sistema, ki zagotavlja ustrezno izvajanje načrtovanih funkcij i

zaúpnost -i ž (*angl. confidentiality*) lastnost sredstva, sistema, ki zagotavlja, da informacije niso nepooblaščno razkrite

MEDNARODNI SIMPOZIJ S PODROČJA OPERACIJSKIH RAZISKAV V SLOVENIJI (SOR '23)

BLED, SLOVENIJA, 20. DO 22. SEPTEMBER 2023

<https://sor.fov.um.si/>

17. mednarodni simpozij s področja operacijskih raziskav v Sloveniji (SOR '23) bo potekal od 20. do 22. septembra 2023 na Bledu v Sloveniji.

Organizirali ga bodo Slovensko društvo Informatika, Sekcija za operacijske raziskave (SDI-SOR, Ljubljana, Slovenija), Univerza v Mariboru, Fakulteta za organizacijske vede (UM-FOV, Kranj, Slovenija), in Univerza v Ljubljani, Fakulteta za strojništvo (UL-FS, Ljubljana, Slovenija).

POMEMBNI DATUMI

Oddaja prispevkov: 2. junij 2023

Poročila recenzentov: 3. julij 2023

Predložitev revidiranih prispevkov: 25. avgust 2023

DRUŽABNI PROGRAM

Sreda, 20. september 2023, popoldne: Druženje z izletom in večerjo.

Četrtek, 21. september 2023, zvečer: Druženje.

PUBLIKACIJE

Natisnjeni Zbornik SOR'23 bo na voljo vsem prijavljenim udeležencem simpozija. Predvidene so posebne številke revij Central European Journal of Operational Research, Business Systems Research Journal, Organizacija in Naše gospodarstvo Our economy, v katerih bodo objavljene revijalne različice izbranih prispevkov.

SOPHOS

Cybersecurity delivered.



Sophos Managed Threat Response

DRUGI SE USTAVIJO SAMO PRI OBVESTILU O GROŽNJI.

**SOPHOS MTR STROKOVNJAKI
GROŽNJO TUDI ODSTRANIJO - 24/7!**

Distributer: Sophos d.o.o., www.sophos.si, slovenija@sophos.si, T: 07/39 35 600

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 v program certificiranja ECDL vključenih že preko 16 milijonov oseb, ki so uspešno opravile preko 80 milijonov izpitov in pridobile ustrezne certificate. V Sloveniji je bilo doslej v program certificiranja ECDL vključenih več kot 18.000 oseb in opravljenih več kot 92.000 izpitov. V Sloveniji sta akreditirana dva izpitna centra ECDL, ki imata izpostave po vsej državi.



Znanstveni prispevki

Patrik Rek, Tina Beranič, Marjan Heričko
ŠTUDIJA PRIČAKOVANJ IN UPORABNOSTI ORODIJ ZA RAZVOJ
APLIKACIJ Z MALO ALI NIČ PROGRAMIRANJA

Kratki znanstveni prispevki

Davor Hafnar, Jure Demšar
ENA IGRA ZA VSE: GENERIRANJE PERSONALIZIRANE VSEBINE
V MOBILNIH IGRAH

Tim Oblak, Peter Peer
OCENJEVANJE KAKOVOSTI PRSTNIH SLEDI Z ANSAMBLI
GLOBOKEGA UČENJA

Andraž Krašovec, Veljko Pejović
VPLIV KOGNITIVNE OBREMENJENOSTI NA VEDENJSKE VZORCE
POSAMEZNIKOV PRI OVERJANJU UPORABNIKOV

Strokovni prispevki

Slavko Žitnik, Karmen Kern Pipan, Miha Jesenko, Dejan Lavbič
PONOVRNO UPORABLJIVE SEMANTIČNE SPLETNE KOMPONENTE:
PRIMER UPORABE ZA INTEROPERABILNOST DIGITALNIH JAVNIH STORITEV

Informacije

IZ ISLOVARJA

ISSN 1318-1882



9 771318 188001