Sveučilište u Zagrebu

Fakultet organizacije i informatike Varaždin

Snježana Križanić

RAZVOJ DIGITALNIH BLIZANACA PROIZVODNIH POSLOVNIH PROCESA

DOKTORSKI RAD

Varaždin, 2025. godina

Sveučilište u Zagrebu

Fakultet organizacije i informatike Varaždin

Snježana Križanić

RAZVOJ DIGITALNIH BLIZANACA PROIZVODNIH POSLOVNIH PROCESA

DOKTORSKI RAD

Mentor: prof. dr. sc. Neven Vrček

Varaždin, 2025. godina

University of Zagreb

Faculty of Organization and Informatics in Varaždin
Postgraduate doctoral study “Information Sciences”

Snježana Križanić

DEVELOPMENT OF DIGITAL TWINS OF PRODUCTION BUSINESS PROCESSES

DOCTORAL DISSERTATION

Supervisor: Full Prof. Neven Vrček, Ph.D.

Varaždin, 2025

Sažetak

Modeliranje poslovnih procesa je jedan od presudnih koraka u upravljanju životnim ciklusom poslovnog procesa. Digitalni blizanac poslovnog procesa je virtualni model poslovnog procesa dizajniran kako bi replicirao ponašanje stvarnog poslovnog procesa. Razlika između digitalnog blizanca i simulacije poslovnog procesa je u tome što za simulaciju ne trebaju podaci u realnom vremenu, dok se digitalni blizanac konstruira temeljem stvarnih podataka u realnom vremenu. Digitalni blizanac se sastoji od fizičkog entiteta u stvarnom svijetu, digitalnog blizanca u programskom obliku i podataka koji povezuju dva navedena elementa. Iako je područje digitalnih blizanaca sve popularnije, digitalni blizanci poslovnih procesa nisu dovoljno znanstveno pokriveni te metodika za razvoj digitalnih blizanaca poslovnih procesa nije oblikovana. Izazov je oblikovati metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa i razviti digitalnog blizanca proizvodnog poslovnog procesa primjenom razvijene metodike.

Ključne riječi: *Digitalni blizanci, Internet stvari, poslovni proces, modeliranje poslovnih procesa*.

Abstract

Business process modeling is one of the most important steps in managing the lifecycle of a business process. A digital twin of a business process is a virtual model of a business process designed to replicate the behavior of a real business process. The difference between a digital twin and a business process simulation is that the simulation does not require real-time data, while a digital twin is created based on real-time data. A digital twin consists of a physical entity in the real world, a digital twin in software form, and data that connects the two elements mentioned above. Although the field of digital twins is becoming popular, digital twins of business processes are not sufficiently scientifically covered, and the methodology for the development of digital twins of business processes has not been designed. The challenge is to design a methodology for the development of digital twins of production business processes, and to develop a digital twin of a production BP using a designed methodology.

Keywords: *Digital twins, Internet of things, business process, business process modeling*.

Extended abstract

Business process modeling is one of the most important steps in managing the lifecycle of business processes. A business process digital twin is a virtual model designed to replicate the behavior of real-world business processes using real-time data. Unlike traditional process simulations, which operate with predefined scenarios and do not require real-time data, digital twins dynamically synchronize with their physical counterparts, enabling real-time monitoring, analysis, and optimization. The digital twin paradigm is distinctive due to its strong integration with real-time data sources, ensuring that the virtual representation remains continuously updated and reflective of the actual process state. By leveraging existing business intelligence, rule-based systems, and AI-driven

decision-making models, digital twins enhance the synergy between business and technology, creating an innovative intellectual asset within enterprises. While digital twins have been widely adopted in industries such as product manufacturing, smart cities, and healthcare, the development of digital twins for business processes, especially for production systems, remains an underexplored research area.

This research addresses the challenge of designing a methodology for the development of business process digital twins in production environments, ensuring their realistic emulation of actual production processes. Specifically, the study investigates how existing business process modeling methodologies can be adapted to support the development of production digital twins while considering industrial requirements and the increasing integration of AI-driven predictive analytics. The primary goal of this research is to propose a methodology for the development of production business process digital twins, extending existing business process modeling frameworks. In this study, a methodology is defined as a structured set of steps that guides the development of a digital twin for a production business process. Additionally, the research focuses on integrating predictive maintenance models into production digital twins using artificial neural networks to forecast failures and optimize equipment maintenance. This transformation elevates digital twins from static process models to intelligent systems capable of detecting potential issues in advance and facilitating timely operational decisions.

The research follows the Design Science Research (DSR) paradigm, structured into three key phases: problem definition, solution development, and verification through a simulation experiment. The first phase identifies gaps in current business process modeling methodologies concerning digital twin development. The second phase develops a structured methodology for guiding the development of production digital twins. The third phase validates the methodology through simulation experiments designed to evaluate the effectiveness of the developed approach in accurately emulating real-world production processes. To validate the methodology, two case studies were selected: drone manufacturing in a Swiss Innovation Park and the simulated production of mobile devices in a smart factory at the faculty of the University of Zagreb. These case studies provide insights into the applicability of the proposed methodology in industrial settings, highlighting the integration of advanced technologies such as 3D printing, robotics, and predictive analytics.

This research makes two primary scientific contributions. The first contribution is the development of a methodology for designing production business process digital twins, which extends existing business process modeling approaches. The second contribution lies in enhancing the existing tools for business process modeling in order to enable the integration of predictive analytics into digital twins. Current BPMN tools lack built-in mechanisms for incorporating machine learning models into process diagrams, requiring manual API integrations.

A simulation experiment was conducted to validate the developed methodology, following three key phases. In the modeling phase, a production business process digital twin was designed using BPMN in Camunda, with Python scripts enabling automated data processing and decision-making. In the experimental phase, the digital twin was tested under various scenarios, simulating real-world production conditions. In the data analysis phase, the simulation results were compared against historical production data to evaluate the accuracy and predictive capability of the digital twin. The experiment demonstrated that the developed digital twin accurately emulated real-world production

processes, supporting real-time decision-making and predictive maintenance. The comparison of simulation results with production business process digital twin showed minimal deviations, confirming the reliability of the proposed methodology.

The proposed methodology provides a structured guide for developing intelligent production business process digital twins, enabling real-time monitoring, predictive maintenance, and operational optimization. The research highlights the need for enhancements in business process modeling tools to ensure their adaptability to AI-driven digital twins. By bridging the gap between business process modeling and digital twin technology, this research contributes to the evolution of Industry 4.0 and the transition toward Industry 5.0, where AI-driven automation and intelligent decision-making become central to production systems.

Mojem ocu,
koji je ovo oduvijek htio za mene.

Zahvala

U ovom trenutku, teško je pronaći riječi kojima bih izrazila zahvalnost svima koji su mi pomogli tijekom ovih godina na doktorskom studiju, koliko radim na doktorskoj disertaciji.

Prije svega želim zahvaliti svom mentoru, prof. dr. sc. Nevenu Vrčeku, bez kojeg sve ovo ne bi bilo moguće. Njegove zasluge su daleko najveće jer je bio najveća podrška, potpora i njegova velika želja za mojim uspjehom je ono što me je dovelo do ovog trenutka. Bez njegovih savjeta, uputa, razmišljanja i vodstva, moj doktorat bi vjerojatno ostao samo neispunjen san. Profesor Neven Vrček je i nevjerojatan oslonac u cijelom mom radu. Uz njega znam da sve što radim, radim dobro. Želim se zahvaliti i prof. dr. sc. Diani Šimić, koja je, uz mog mentora, bila sljedeća osoba koja je vjerovala u mene i pomagala je svojim savjetima. Njezino znanje u području znanstveno-istraživačkog rada je nemjerljivo, a vrijeme koje je ona uložila da pomogne je bogatstvo koje vrlo cijenim. Nadalje, želim se zahvaliti prof. dr. sc. Vjeranu Strahonji na savjetima, pomoći i podršci koje je upućivao tijekom vremena pripreme i izrade doktorskog rada. Želim se zahvaliti i prof. dr. sc. Katarini Tomičić-Pupek, koja je u pravom trenutku znala procijeniti što je najbolje za mene. Ja znam da njezine namjere često idu u smjeru mog interesa i dobra.

Ovim putem željela bih se zahvaliti kolegama s Fakulteta strojarstva i brodogradnje u Zagrebu, naročito Matiji Golecu, mag. ing. mech., na dostupnim podacima za izradu doktorata i posvećenom vremenu tijekom višestrukih dolazaka u Pametnu tvornicu na Fakultetu.

Ovo istraživanje ne bi bilo stvarno bez potpore, podrške, ljubavi i motivacije koju mi je stalno pružao moj najveći životni prijatelj, a to je moj suprug, Krešimir. Ovaj doktorat je naš zajednički uspjeh, nakon dvoje prekrasne djece, treći u nizu u životu. Krešimire, hvala ti za to što jesi.

Za kraj, želim se zahvaliti svim prijateljima, kolegama s Fakulteta organizacije i informatike u Varaždinu na svoj podršci i savjetima koje su pružali tijekom ovih godina na doktorskom studiju. Stvari se čine lakšima kada nisi sam.

I najteži put postaje lakši kada ga ne hodaš sam jer snaga ne leži samo u koracima, nego i u ruci koja te prati.

Životopis mentora

Dr. sc. Neven Vrček redoviti je profesor u trajnom zvanju na Fakultetu organizacije i informatike, Sveučilišta u Zagrebu (FOI). Diplomirao je na Elektrotehničkom fakultetu (danas FER) Sveučilišta u Zagrebu 1991. godine. Na istom je fakultetu magistrirao u listopadu 1995. godine te obranio doktorski rad u prosincu 1998. Profesionalno obrazovanje nadopunjavao je nizom seminara i studijskih boravaka na fakultetima i poduzećima (University of Kentucky - School of Business and Economics, SAD, Mannesmann-Kienzle, edukacijski centar Donnau Eschingenn, ORACLE, edukacijski centar Varšava, University of Georgia, Institute od Higher Education, SAD). Ovo je obrazovanje uglavnom bilo vezano uz modernu organizaciju poduzeća i primjenu informacijskih i komunikacijskih tehnologija u poslovanju što mu je trajan predmet znanstvenog i stručnog interesa.
Na FOI-ju izvodi nastavu na više kolegija na svim razinama studija prvenstveno u domeni primjene informacijske tehnologije u organizacijama javnog i privatnog sektora – Digitalno poslovanje, Elektronička uprava, Modeliranje poslovnih procesa, Strateško planiranje razvoja informacijskih sustava. U periodu 2014.-2018. bio je voditelj poslijediplomskog interdisciplinarnog specijalističkog studija Javna uprava na Sveučilištu u Zagrebu na kojem izvodi nastavu iz kolegija Elektronička potpora javnoj upravi. Na specijalističkim poslijediplomskim studijima Računovodstvo i porezi te Financijsko izvještavanje, revizija i analiza, Ekonomskog fakulteta Sveučilišta u Zagrebu predaje kolegij „Informacijske tehnologije u računovodstvu“. Autor je većeg broja znanstvenih i stručnih radova. Dio znanstvene djelatnosti ostvaruje preko znanstvenih projekata te je sudjelovao i vodio više projekata financiranih iz domaćih i EU izvora. Bio je višegodišnji urednik znanstvenog časopisa *Journal of Information and Organizational Sciences* (JIOS) te predsjednik programskog odbora konferencije *Central European Conference of Information and Intelligent Systems* (CECIIS). Član je programskog odbora više međunarodnih znanstvenih konferencija od kojih su istaknutije: ISD – *Information Systems Development* te *International Conference on Mobility and Smart Cities*. Član je savjetodavnog uređivačkog odbora časopisa *Annals of Management Science* te uređivačkog odbora časopisa *Social Technologies*. Mentorirao je 15 doktorata znanosti i 10 magisterija znanosti.

Na FOI-ju je deset godina bio član uprave (prodekan i dekan). Također je deset godina bio i voditelj doktorskog studija Informacijske znanosti. Trenutno je voditelj poslijediplomskog doktorskog studija Upravljanje digitalnim inovacijama koji zajednički izvode Fakultet organizacije i informatike te Ekonomski fakultet Sveučilišta u Zagrebu. Pored znanstvenog rada aktivan je i stručno te povremeno radi kao konzultant za razvoj poslovne izvrsnosti i strateško planiranje upotrebe informacijskih tehnologija u više poduzeća te javnih ili vladinih ustanova. U periodu od 2006. do 2009. bio je član Nacionalnog vijeća za informacijsko društvo RH (odlukom Vlade Republike Hrvatske od 20. travnja 2006. godine). Bio je član Radne skupine Vlade RH za izradu i provedbu strategije elektroničkog poslovanja (od lipnja 2006. do kraja rada u kolovozu 2007. godine) te član radne skupine Vlade RH za razvoj Strategije digitalnog rasta. Član je europske mreže eksperata za otvorene podatke javnog sektora i semantičku interoperabilnost - Share-PSI (http://www.w3.org/2013/share-psi/). Predsjednik je Revizorskog odbora Hrvatske narodne banke.

Za uspješan rad dobio je više društvenih priznanja, kao npr. srebrnu značku (2008.) Hrvatskog informatičkog zbora, medalju grada Varaždina (2007.), Memorial Eurocoin of Aurel Stodola. Živi i radi u Varaždinu. Oženjen je i otac dvoje djece.

Sadržaj

Popis kratica

API	Aplikacijsko programsko sučelje (eng. *Application Programming Interface*)
BPM	Upravljanje poslovnim procesima (eng., *Business Process Management*)
BPMN	Notacija za modeliranje poslovnih procesa (eng. *Business Process Model and Notation*)
BPDT	Digitalni blizanac poslovnog procesa (eng. *Business Process Digital Twin*)
CPS	Kibernetičko – fizički sustav (eng. *Cyber – Physical System*)
CSV	Vrijednosti odvojene zarezima (eng. *Comma-Separated Values*)
DB	Digitalni blizanac
DSR	Znanstveno istraživanje dizajna (eng. *Design Science Research*)
ERP	Sustav za planiranje resursa poduzeća (eng. *Enterprise Resource Planning*)
FSB	Fakultet strojarstva i brodogradnje
IIoT	Industrijski Internet stvari (eng. *Industrial Internet of Things*)
IoT	Internet stvari (eng. *Internet of Things*)
JSON	Notacija objekata u JavaScriptu (eng. *JavaScript Object Notation*)
MLP	Višeslojni perceptron (eng. *Multilayer Perceptron*)
RFID	Identifikacija radio-frekvencijom (eng. *Radio Frequency Identification*)
RNN	Rekurentna neuronska mreža (eng. *Recurrent Neural Network*)
SCADA	Nadzorno upravljanje i prikupljanje podataka (eng. *Supervisory Control and Data Acquisition*)
SOA	Arhitektura usmjerena na servise (eng. *Service-Oriented Architecture*)
SQL	Strukturni upitni jezik (eng. *Structured Query Language*)
UML	Unificirani jezik za modeliranje (eng. *Unified Modeling Language*)

1 Uvod

Digitalni blizanci sve više zauzimaju važno mjesto u modernim industrijskim i poslovnim sustavima zahvaljujući svojoj sposobnosti da povežu fizički i digitalni svijet u jedinstvenu cjelinu. Postoji puno definicija digitalnog blizanca u literaturi i one su prilagođene različitim kontekstima primjene. U ovom istraživanju definicija digitalnog blizanca je da je digitalni blizanac virtualna reprezentacija objekta ili sustava, koja je obogaćena skupom stvarnih podataka iz realnog vremena povezanih s tim objektom ili sustavom, te omogućuje precizno predviđanje performansi fizičkog entiteta tijekom vremena. Paradigma digitalnog blizanca specifična je zbog važnosti povezivanja digitalnog blizanca s izvorima stvarnih podataka iz okruženja i ažuriranja njegovog stanja u stvarnom vremenu (Križanić, S. & Vrček, N., 2023). Digitalni blizanac potiče sinergijsku povezanost između poslovanja i tehnologije korištenjem postojeće poslovne inteligencije, pravila i modela umjetne inteligencije temeljenih na informacijskim tehnologijama. Digitalni blizanac pruža poseban mehanizam koji stvara novo intelektualno vlasništvo unutar poduzeća, čineći ga važnim i jedinstvenim resursom za organizacije usmjerene na budućnost (Križanić, S. & Vrček, N., 2024). Tako digitalni blizanci postaju ključna tehnologija u modernim industrijskim sustavima. Oni omogućavaju detaljno modeliranje, analizu i optimizaciju poslovnih procesa.

Unatoč njihovoj širokoj primjeni u različitim područjima i industrijama, od simulacije proizvoda i uređaja do pametnih gradova, nedovoljno je istraženo kako razviti digitalnog blizanca proizvodnog poslovnog procesa i kako osigurati njegovu realističnu emulaciju stvarnog proizvodnog okruženja. Posebno se postavlja pitanje kako postojeće metodike modeliranja poslovnih procesa mogu podržati razvoj digitalnih blizanaca proizvodnih poslovnih procesa, uzimajući u obzir zahtjeve industrijske proizvodnje i sve veću integraciju umjetne inteligencije u optimizaciju poslovnih odluka. Ovo istraživanje bavi se upravo tim istraživačkim izazovima.

Poslovni proces predstavlja skup međusobno povezanih aktivnosti koje se provode radi postizanja unaprijed definiranog cilja, pri čemu se koriste resursi organizacije kako bi se proizvele vrijednosti za korisnike. Ti procesi uključuju ulazne elemente, niz transformacijskih aktivnosti i izlazne rezultate koji zadovoljavaju zahtjeve korisnika i/ili tržišta. Proizvodni poslovni proces je skup međusobno povezanih aktivnosti koje poduzeća koriste kako bi ekonomske ulazne resurse, poput rada, opreme ili zemljišta, pretvorila u proizvode i usluge za potrošače. Ovaj proces usmjeren je na učinkovitu proizvodnju proizvoda i njihovu brzu isporuku kupcima, uz istovremeno očuvanje kvalitete proizvoda (Indeed Editorial Team, 2025). Proizvodni poslovni proces razlikuje se od neproizvodnih poslovnih procesa po tome što uključuje fizičku transformaciju sirovina u gotove proizvode, dok neproizvodni procesi obuhvaćaju uslužne, administrativne ili informacijske aktivnosti bez materijalne proizvodnje. Proizvodni poslovni procesi korišteni u ovom istraživanju odlikuju se primjenom suvremenih tehnologija i informacijsko-komunikacijskih rješenja (ICT) u svakodnevnom poslovanju. Izvođenje procesa je standardizirano i jasno definirano, pri čemu aktivnosti slijede unaprijed utvrđeni redoslijed. Procesi se odvijaju deterministički, bez značajnih elemenata slučajnosti, što omogućuje njihovu preciznu analizu i dosljednu ponovljivost.

Ovo istraživanje predlaže metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa proširenjem postojećih metodika modeliranja poslovnih procesa. U ovom istraživanju metodika je skup koraka koji objašnjava kako razviti proizvodnog digitalnog blizanca. Istraživanje se bavi i

integracijom modela prediktivnog održavanja u proizvodne digitalne blizance, pri čemu se koriste neuronske mreže za predviđanje kvarova i optimizaciju održavanja opreme. Neuronske mreže nisu predmet detaljne analize u ovom istraživanju, već se koriste isključivo kao način kako implementirati prediktivno održavanje unutar proizvodnih digitalnih blizanaca. Područje umjetne inteligencije obuhvaća znatno širi spektar metoda i pristupa koji nisu obuhvaćeni ovim radom. Fokus istraživanja je na razvoju metodike za proizvodne digitalne blizance, dok se primjena neuronskih mreža razmatra u funkcionalnom kontekstu prediktivne analitike. Digitalni blizanci time postaju ne samo statički modeli procesa, već inteligentni sustavi koji mogu unaprijed detektirati potencijalne probleme u proizvodnji i omogućiti donošenje pravovremenih operativnih odluka.

Istraživanje se temelji na paradigmi znanstvenog istraživanja dizajna (eng., ***D**esign **S**cience **R**esearch*) i provodi se kroz tri ključne faze: definiranje problema, razvoj rješenja i verifikaciju metodike kroz simulacijski eksperiment. Kroz ove faze razvijen je digitalni blizanac proizvodnog poslovnog procesa koji realno emulira stvarni proizvodni sustav. Verifikacija se provodi usporedbom simuliranih rezultata s povijesnim podacima te analizom izvedbe digitalnog blizanca u simuliranom okruženju.

U svrhu provjere metodike odabrana su dva slučaja korištenja: proizvodnja dronova u švicarskom Inovacijskom parku i simulirana proizvodnja mobilnih uređaja u Pametnoj tvornici. Ovi slučajevi korištenja omogućuju analizu primjenjivosti metodike u industriji, s naglaskom na integraciju naprednih tehnologija poput 3D printanja i robotike.

Ovo istraživanje ima dva znanstvena doprinosa. Prvi se ogleda u oblikovanju metodike za razvoj digitalnih blizanaca proizvodnih poslovnih procesa, koja se temelji na proširenju postojećih metodika modeliranja poslovnih procesa. Drugi znanstveni doprinos je taj što istraživanje predlaže poboljšanje alata za modeliranje poslovnih procesa kako bi se omogućila integracija prediktivne analitike u proizvodne digitalne blizance.

2 Upravljanje poslovnim procesima i alati za modeliranje poslovnih procesa

Cilj ovog poglavlja je pružiti sustavan pregled suvremenih standarda i alata koji se koriste za modeliranje i upravljanje poslovnim procesima, uz naglasak na njihovu ulogu u unaprjeđenju učinkovitosti, fleksibilnosti i interoperabilnosti organizacijskih sustava. Kroz analizu relevantnih znanstvenih radova poglavlje nastoji prikazati kako se različiti alati razvijaju i usavršavaju u skladu s potrebama organizacija, uključujući web-bazirane platforme za fleksibilnije modeliranje, podršku odlučivanju pri odabiru alata, upravljanje rizicima te integraciju naprednih tehnologija poput analize sentimenta, prediktivnih algoritama i automatizacije procesa. Poseban naglasak stavljen je na razliku između standarda, koji definiraju formalna pravila i notaciju modeliranja, i alata, koji omogućuju njihovu konkretnu primjenu kroz vizualno modeliranje, simulaciju i analizu. Poglavlje također analizira evaluacijske okvire koji omogućuju sustavni pristup odabiru BPM alata na temelju tehničkih, funkcionalnih i organizacijskih kriterija. Analizom prednosti i ograničenja različitih pristupa, uključujući aspektno orijentirane alate, hibridne metode odlučivanja i rješenja za upravljanje rizicima, poglavlje pokazuje kako alati za modeliranje poslovnih procesa evoluiraju kako bi odgovorili na izazove digitalne transformacije uključujući razvoj digitalnih blizanaca poslovnih procesa u proizvodnom kontekstu..

Standard za modeliranje poslovnih procesa definira koncepte, notaciju i pravila za vizualno predstavljanje poslovnih aktivnosti, tokova podataka i odluka unutar organizacije. Najčešće korišteni standardi su BPMN (eng., *Business Process Model and Notation*), koji omogućuje detaljno i razumljivo modeliranje procesa (IBM, 2025b) te UML (eng., *Unified Modeling Language*) za šire softversko inženjerstvo.

Autori (Arnold, L. i ostali, 2025) predstavljaju Web-bazirani alat za modeliranje poslovnih procesa, nazvan *PHILharmonicFlows*. Cilj istraživanja bio je razviti poboljšanu verziju prethodno postojećeg lokalnog modeliranja, omogućujući fleksibilniju, dostupniju i korisnički pristupačniju platformu. Alat je razvijen kao Web-aplikacija koja omogućuje automatizaciju životnih ciklusa objekata. Implementacija uključuje napredne verifikacijske algoritme, metričke sustave za praćenje te fleksibilnije definiranje poslovnih pravila i ograničenja. Rezultati istraživanja pokazuju da Web-bazirana verzija alata olakšava modeliranje kompleksnih poslovnih procesa, smanjuje rizik od grešaka i povećava učinkovitost kroz automatizirano predviđanje procesa pomoću, tzv., Kalmanovog filtra.

U radu (Iizuka, K. i ostali, 2014) autori istražuju izazove prilagodbe međuorganizacijskih poslovnih procesa i predlažu alat za prilagodbu toka međuorganizacijskih procesa (IPFA alat). Autori analiziraju radove o modeliranju poslovnih procesa iz perspektive modeliranja i procesa prilagodbe, budući da to može doprinijeti rješavanju problema modeliranja u poboljšanju i reinženjeringu poslovnih procesa japanskih poduzeća, posebno proizvodnih tvrtki koje su svjesne potrebe za transformacijom poslovnih procesa, ali je još nisu ostvarile. Cilj istraživanja bio je razviti alat koji podržava usklađivanje poslovnih procesa između različitih organizacija, uzimajući u obzir japanske specifičnosti poput sklonosti prilagođenim softverskim rješenjima. Alat omogućuje višeslojno prikazivanje procesa, gdje korisnici mogu analizirati i prilagođavati tokove podataka kroz različite

organizacijske razine. Evaluacija je provedena na primjeru sveučilišnog administrativnog procesa, pri čemu su korisnici mogli jednostavnije pratiti, analizirati i prilagoditi poslovne procese kroz intuitivno sučelje. Rezultati pokazuju da alat olakšava komunikaciju između operativnih i menadžerskih razina te poboljšava prilagodbu poslovnih procesa u složenim organizacijskim strukturama.

Autori (Jin, G. i ostali, 2022) istražuju problem odabira alata za modeliranje poslovnih procesa koristeći hibridnu metodu *Fuzzy DEMATEL* i *TOPSIS*. Cilj istraživanja bio je razviti višekriterijski model odlučivanja koji pomaže organizacijama u odabiru optimalnog alata za modeliranje poslovnih procesa, uzimajući u obzir tehničke, ekonomske, vremenske i sigurnosne parametre. Istraživanje je pokazalo da se integrirani pristup može učinkovito koristiti za donošenje odluka o odabiru alata, pri čemu su rezultati testirani na osam kandidata, uključujući Bizagi Modeler, ADONIS:CE i Cardanit.

Istraživanje (Lüftenegger, E. & Softic, S., 2020) predstavlja SentiProMo, alat za modeliranje poslovnih procesa koji koristi analizu sentimenta za poboljšanje upravljanja poslovnim procesima. Cilj istraživanja bio je razviti rješenje koje omogućuje integraciju ideja sudionika u analizu i redizajn poslovnih procesa, pružajući analitičarima uvid u pozitivne i negativne aspekte trenutnih procesa. Alat omogućuje unos komentara korisnika na razini zadataka unutar procesa, primjenjuje algoritme obrade prirodnog jezika za analizu sentimenta te vizualizira rezultate kroz pozitivne i negativne ocjene. Evaluacija je provedena na primjeru procesa ukrcaja putnika u zračnoj luci, pri čemu su komentari sudionika korišteni za identifikaciju problematičnih točaka i mogućnosti poboljšanja. Rezultati pokazuju da se analiza sentimenta može uspješno koristiti za unapređenje dizajna poslovnih procesa.

Autori (Medeiros, A.L. i ostali, 2017) istražuju kako proširiti alat Oryx u aspektno-orijentirani alat za modeliranje poslovnih procesa, nazvan CrossOryx. Cilj istraživanja bio je unaprijediti modularnost modela poslovnih procesa, čime bi se omogućilo bolje upravljanje složenim i međusobno povezanim procesima. Autori su razvili CrossOryx alat proširujući postojeći Oryx, dodajući nove elemente za modeliranje. Validacija alata provedena je kroz usporedbu s prilagođenom verzijom ARIS alata, pri čemu autori ističu da je CrossOryx pokazao bolje rezultate u pogledu fleksibilnosti i modularnosti modela.

U radu (Padmanabhuni, S. i ostali, 2004) se istražuje komplementarnost Web servisa, grid računarstva i upravljanja poslovnim procesima s ciljem povećanja poslovne agilnosti. Nadalje, cilj istraživanja bio je analizirati kako ove tehnologije mogu zajedno omogućiti fleksibilnije i prilagodljivije poslovne procese, smanjujući troškove i poboljšavajući interoperabilnost IT sustava. Autori tako istražuju kako se BPM nadovezuje na konvencionalne koncepte tijekova rada i proširuje ih zamjenom fiksnih veza u tijekovima rada parametriziranim i konfigurabilnim vezama. Ova parametrizacija i konfigurabilnost ključni su za postizanje fleksibilnosti IT arhitekture i poslovnih procesa. Istraživanje predlaže korištenje *Open Grid Services Architecture* (OGSA) i BPEL4WS za orkestraciju Web servisa unutar grid sustava te analizira primjenu u financijskom sektoru, posebice u analizi rizika.

Autori (Popa, C. & Goia, I., 2024) istražuju primjenu alata za modeliranje poslovnih procesa u optimizaciji operacija prekrcaja tereta. Cilj istraživanja bio je modelirati i analizirati prekrcajne procese kako bi se identificirale mogućnosti za optimizaciju i poboljšanje učinkovitosti. Korišten je alat Aura Portal Modeler za simulaciju i analizu poslovnih procesa, pri čemu su identificirana uska

grla i predložene mjere poboljšanja, uključujući praćenje zaliha u stvarnom vremenu, optimizaciju rasporeda skladištenja i analizu učinkovitosti opreme. Rezultati simulacije ukazuju na to da se primjenom optimizacijskih strategija može smanjiti vrijeme iskrcaja, poboljšati raspodjela resursa i smanjiti operativni troškovi. Daljnja istraživanja usmjerena su na preciznije modeliranje varijabilnosti procesa i integraciju naprednih tehnologija poput automatizacije i prediktivne analitike za predviđanje potražnje i planiranje kapaciteta.

U radu (Reijers, H.A. i ostali, 2011) se istražuje primjena sintaksnog isticanja (eng., *syntax highlighting*) u modeliranju poslovnih procesa radi poboljšanja razumljivosti modela. Cilj istraživanja bio je ispitati kako „bojanje“ sintaktičkih elemenata u modelima poslovnih procesa može poboljšati njihovu čitljivost i olakšati korisnicima razumijevanje odnosa među elementima. Autori su koristili alat WoPeD, koji je otvoreno softversko rješenje za modeliranje poslovnih procesa . Dodatno su testirali učinke sintaksnog isticanja na razumijevanje modela među korisnicima različite razine stručnosti. Rezultati pokazuju da isticanje poboljšava točnost razumijevanja modela, posebno kod manje iskusnih korisnika, dok utjecaj na brzinu razumijevanja nije bio statistički značajan.

Autori (Tang, W. i ostali, 2009) predstavljaju *CuteFlow*, alat za modeliranje poslovnih procesa temeljen na unaprijeđenom meta-modelu *CuteMeta*. Cilj istraživanja bio je razviti fleksibilan alat za modeliranje poslovnih procesa koji omogućuje lakšu interoperabilnost između različitih BPM meta-modela te podršku za *Service-Oriented Architecture* (SOA) i *Model-Driven Architecture* (MDA) paradigme. Autori su analizirali četiri dominantna BPM meta-modela (BPMN, UML *Activity Diagram*, XPDL i BPEL) te su iz njih izvukli zajedničke semantičke elemente kako bi definirali pojednostavljen i proširiv meta-model *CuteMeta*. *CuteFlow* je tako razvijen kao grafički BPM alat. Validacija je pokazala da *CuteFlow* omogućuje učinkovitu konverziju poslovnih procesa među heterogenim modelima, poboljšava vizualizaciju i rješava probleme sinkronizacije uz podršku za izravno uređivanje izvornog koda..

Istraživanje (Thabet, R. i ostali, 2018) predstavlja ADOxx, alat za modeliranje poslovnih procesa s integriranim upravljanjem rizicima. Autori uvode metodu modeliranja koju nazivaju ADoBPRIM. Navedeni alat omogućuje dizajn poslovnih procesa vođen rizicima. Cilj istraživanja bio je razviti metodu modeliranja poslovnih procesa koja omogućuje dinamičku procjenu rizika unutar poslovnih modela, povezujući tradicionalno odvojene discipline upravljanja poslovnim procesima i upravljanja rizicima. Autori se oslanjaju na metodu *Business Process-Risk Integrated Method* (BPRIM), koja kombinira modeliranje procesa s analizom rizika, proširujući postojeće poslovne notacije s novim konstrukcijama za modeliranje rizika. Implementacija je izvedena pomoću ADOxx platforme, koja omogućuje vizualno modeliranje, provjeru modela i analizu rizika. Alat su testirali na studiji slučaja sustava upravljanja lijekovima u zdravstvenim ustanovama, pri čemu je pokazano da modeliranje rizika omogućuje bolje razumijevanje mogućih prijetnji i poboljšanje sigurnosti procesa. Slično su napravili i u radu (Thabet, R. i ostali, 2021) gdje također prikazuju kako su razvili *ADoBPRIM*. U ovom radu autori su se oslonili na e-BPRIM, digitaliziranu verziju BPRIM metode, koja omogućuje modeliranje rizika unutar poslovnih procesa pomoću ADOxx platforme. Alat su testirali na tri zdravstvena procesa u Francuskoj, uključujući proces sterilizacije medicinskih uređaja i upravljanje rizicima u procesu primjene lijekova.

Autori (Tomaz, L.F.C. i ostali, 2011) istražuju kako poboljšati sustav preporuka unutar alata za modeliranje poslovnih procesa *Business Process Cooperative Editor* (BPCE) primjenom koncepta

Knowledge Vectors (KV). Cilj istraživanja bio je unaprijediti algoritam preporuka u BPCE-u kako bi bolje odražavao razinu znanja korisnika o predmetnom modelu. Rezultati su pokazali da uvođenje koncepta *Knowledge Vectors* poboljšava točnost preporuka iz razloga što modeli koje razvijaju stručnjaci u danom području dobivaju višu ocjenu i imaju veću vjerojatnost da budu prepoznati kao najbolji.

Autori (Walterbusch, M. i ostali, 2013) predlažu okvir kriterija za evaluaciju alata za modeliranje poslovnih procesa, temeljen na studijama slučaja i sustavnom pregledu literature. Cilj istraživanja bio je razviti strukturiran pristup za odabir BPM alata koji uzima u obzir širok spektar tehničkih, funkcionalnih i organizacijskih kriterija. Autori su proveli sustavni pregled literature kako bi identificirali relevantne kriterije te su razvili hijerarhijski model evaluacije koji sadrži više od 950 kriterija, organiziranih u sedam razina. Tih sedam razina su:

1. Jezici modeliranja – analiza broja i tipa modelirajućih jezika koje alat podržava (npr. BPMN, EPC, UML).
2. Izvršenje procesa – sposobnost alata da modelirane procese pokrene i izvrši u stvarnim okruženjima.
3. Praćenje troškova procesa – mogućnosti alata za procjenu troškova povezanih s procesima.
4. Validacija sintakse procesa – provjera ispravnosti modela u skladu s definiranim pravilima.
5. Funkcionalnost ispisa i izvoza modela – podržava li alat različite formate izvoza i vizualizacije modela.
6. Troškovi i licenciranje – analiza troškova povezanih s korištenjem alata, uključujući modele licence.
7. Dokumentacija i korisnička podrška – kvaliteta i dostupnost korisničkih priručnika te broj postojećih implementacija.

Okvir je validiran kroz studiju slučaja u sektoru zdravstvene skrbi, gdje je korišten za odabir alata prikladnog za modeliranje procesa skrbi o pacijentima. Rezultati pokazuju da predloženi okvir omogućuje precizniji i sustavniji odabir BPM alata te poboljšava transparentnost odlučivanja.

Autori (zur Muehlen, Michael & Rosemann, Michael, 2000) istražuju praćenje i kontrolu poslovnih procesa temeljenih na radnim tokovima s tehničkog i organizacijskog aspekta. Cilj istraživanja bio je razviti metodiku i alat za analizu podataka iz sustava kako bi se poboljšala vidljivost i učinkovitost poslovnih procesa. Autori su unutar istraživačkog projekta CONGO razvili alat PISA (eng., *Process Information System for Analysis*), koji omogućuje analizu podataka iz različitih *workflow* sustava u kombinaciji s podacima iz alata za modeliranje poslovnih procesa. Nadzor i upravljanje procesima temeljenim na tijeku rada omogućuje detaljniju evaluaciju poslovnih procesa zahvaljujući poboljšanoj kvaliteti podataka. To predstavlja značajnu priliku za razvoj sustava za upravljanje informacijama. Također omogućuje provođenje analiza koje su prethodno bile nemoguće zbog nedostatka kvalitetnih podataka.

Standardi za modeliranje poslovnih procesa i alati za modeliranje poslovnih procesa često se koriste zajedno, no ključna je razlika između njih. Standardi poput BPMN-a, UML dijagrama i BPEL-a definiraju pravila, simbole i sintaksu koja omogućuje jednoznačno predstavljanje poslovnih procesa, neovisno o alatu koji se koristi. S druge strane, BPM alati poput Bizagi Modelera, ARIS Toolseta i drugih, implementiraju te standarde, omogućujući vizualizaciju, simulaciju, analizu i optimizaciju poslovnih procesa. Radovi analizirani u ovom poglavlju pokazuju da razvoj BPM alata ide u smjeru

povećane pristupačnosti, podrške upravljanju rizicima (Thabet, R. i ostali, 2021), integracije analize sentimenta (Lüftenegger, E. & Softic, S., 2020) te boljeg donošenja odluka pri odabiru alata (Jin, G. i ostali, 2022). Nadalje, BPM alati postaju sve sofisticiraniji, povezujući različite paradigme poput *Service-Oriented Architecture* (SOA), *Model-Driven Architecture* (MDA) i aspektnog modeliranja (Medeiros, A.L. i ostali, 2017), (Tang, W. i ostali, 2009). Iako su standardi nužni za interoperabilnost i dosljednost modela, alati donose dodatne funkcionalnosti, poput izvršenja procesa, analize podataka i podrške za donošenje odluka. Tako kažemo da alati proširuju standarde za svoje potrebe.

Drugim riječima, standard i alati za modeliranje poslovnih procesa nisu isto jer standard definira pravila, notaciju i sintaksu, dok alati omogućuju praktičnu primjenu tog standarda kroz vizualno modeliranje, analizu i optimizaciju procesa. Standard osigurava jedinstven način prikaza poslovnih procesa, čime se omogućuje interoperabilnost između različitih alata i organizacija. S druge strane, alati pružaju korisničko sučelje i funkcionalnosti koje omogućuju korisnicima da modeliraju procese, provode simulacije, analiziraju performanse i integriraju modele s drugim IT sustavima. Ključna razlika je u tome što standard ne obavlja modeliranje sam po sebi, on samo definira pravila kako bi se osigurala dosljednost i razumljivost modela, dok alati omogućuju njihovu konkretnu implementaciju i primjenu u poslovnom okruženju. Stoga je standard ono što uglavnom ne zahtijeva promjene, međutim, alati za modeliranje poslovnih procesa su podložni promjenama te se tako javlja potreba da se postojeći alati za modeliranje poslovnih procesa prilagode da podržavaju razvoj digitalnih blizanaca proizvodnih poslovnih procesa.

Analiza suvremenih standarda i alata za modeliranje poslovnih procesa pokazuje značajnu evoluciju prema web-baziranim rješenjima, integraciji naprednih tehnologija poput analize sentimenta i prediktivnih algoritama, te proširenju funkcionalnosti na upravljanje rizicima i podršku odlučivanju. Ključna razlika između standarda i alata jasno je definirana: standardi poput BPMN-a osiguravaju jedinstveni sustav zapisivanja poslovnih procesa i interoperabilnost, dok alati omogućuju praktičnu primjenu kroz vizualno modeliranje, simulaciju i analizu. Razvoj evaluacijskih okvira i hibridnih metoda za odabir alata pokazuje potrebu za sustavnim pristupom koji uzima u obzir tehničke, ekonomske i organizacijske parametre. Tendencija prema integraciji različitih paradigmi, uključujući SOA, MDA i aspektno modeliranje, omogućuje veću modularnost i fleksibilnost poslovnih procesa.

U kontekstu digitalne transformacije proizvodnih organizacija, analizirani alati pružaju važnu osnovu za razvoj digitalnih blizanaca proizvodnih poslovnih procesa. Web-bazirane platforme s mogućnostima automatizacije i prediktivne analize predstavljaju korak prema inteligentnim sustavima za modeliranje kompleksnih proizvodnih procesa u virtualnom okruženju. Analiza dostupnih radova jasno ukazuje na potrebu za razvojem alata koji bolje podržavaju razvoj i upravljanje digitalnim blizancima proizvodnih poslovnih procesa kroz integraciju s IoT senzorima, podacima u stvarnom vremenu i naprednim analitičkim tehnikama, što predstavlja temelj za istraživanje provedeno u ovom radu.

3 BPM, BPMN i digitalni blizanci

Upravljanje poslovnim procesima (eng., *Business Process Management*) je disciplina koja se bavi modeliranjem, analizom, unapređenjem i automatizacijom poslovnih procesa radi povećanja organizacijske učinkovitosti. Posljednjih nekoliko godina integracija BPM-a s novim tehnologijama, poput digitalnih blizanaca, otvora nove mogućnosti za poboljšanje procesa i donošenje odluka. Digitalni blizanci, koji su u početku ponajviše bili povezani s proizvodnjom i fizičkim objektima, sve se više primjenjuju u upravljanju poslovnim procesima kroz koncept digitalnih blizanaca poslovnih procesa (eng., *Business Process Digital Twins* – BPDT). U ovom poglavlju će biti prikazana integracija BPM-a, BPM notacije (BPMN) i digitalnih blizanaca, oslanjajući se na rezultate postojeće literature i ističući doprinose autora iz ovog područja.

3.1 Upravljanje poslovnim procesima i digitalni blizanci poslovnih procesa

Implementacija digitalnih blizanaca u poslovnim procesima predstavlja značajan pomak u odnosu na tradicionalno upravljanje procesima. Abouzid i Saidi (2023) predstavljaju strukturirani pristup implementaciji digitalnih blizanaca u procesima opskrbnog lanca, naglašavajući njihovu sposobnost za praćenje u stvarnom vremenu, simulaciju i optimizaciju. Njihov rad pruža temeljno razumijevanje kako se digitalni blizanci mogu integrirati u poslovne operacije, usklađujući se s načelima BPM-a radi povećanja učinkovitosti procesa.

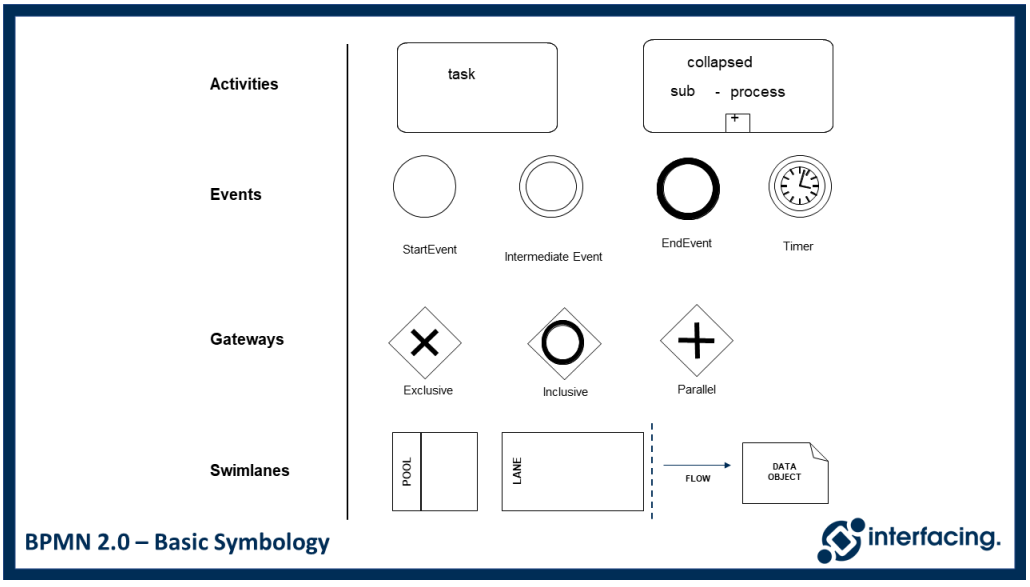
Sumereder i Woitsch (2022) istražuju principe digitalizacije potrebne za razvoj digitalnih blizanaca na razini organizacije. Oni tvrde da digitalni blizanci nadilaze pojedinačne procese i obuhvaćaju cjelovit pregled organizacijskih aktivnosti, čineći ih ključnim dijelom strategija poslovne transformacije. Njihovo istraživanje naglašava važnost usklađivanja BPM praksi s tehnologijom digitalnih blizanaca kako bi se stvorili tzv. digitalni blizanci organizacija (DTOs). Autori govore o korištenju BPMN-a za modeliranje poslovnih procesa koji se zatim u stvarnom vremenu preslikavaju u digitalne blizance. Nadalje, napravili su prikaz dizajniranja i razvoja fizičkih eksperimenata za povezivanje zahtjeva slučaja upotrebe i eksperimentalnih zahtjeva s obrascima digitalizacije prema digitalnim blizancima.

Corradini i sur. (2023) uvode koncept izvršnih digitalnih blizanaca procesa (eng., *Executable Digital Process Twins*) te se zalažu za unapređenje sustava vođenih procesima kombiniranjem BPMN modela s izvršnim digitalnim blizancima. Njihov rad naglašava potencijal digitalnih blizanaca za praćenje i automatizaciju poslovnih procesa u stvarnom vremenu, pružajući razinu transparentnosti i prilagodljivosti koju tradicionalni BPM sustavi ne mogu postići. Autori prikazuju koncepte i tehnologije koje podržavaju primjenu izvršnih digitalnih blizanaca procesa na uređaje Interneta stvari i robotske sustave te predstavljaju implementaciju i primjenu izvršnih digitalnih blizanaca procesa u scenariju s više robota.

3.2 BPMN kao osnova za digitalne blizance proizvodnih poslovnih procesa

Prema Torresu i sur. (2024) BPMN ima ključnu ulogu u prevladavanju prepreka digitalizaciji u upravljanju gradilištima. Autori pokazuju kako digitalni blizanci temeljeni na BPMN-u pomažu u automatizaciji i koordinaciji aktivnosti na gradilištu, omogućujući prilagodbe i donošenje odluka u stvarnom vremenu. Primjena tehnologije digitalnih blizanaca može omogućiti prijelaz s upravljanja projektiranjem kojeg vode isključivo ljudi na hibridni model, gdje surađuju ljudi i digitalni sustavi, kombinirajući stručna znanja s metodama umjetne inteligencije. Nadalje, autori naglašavaju potrebu za standardiziranim procesnim modelima kako bi se osigurala interoperabilnost i skalabilnost pri implementaciji digitalnih blizanaca.
Mallek-Daclin i sur. (2023) istražuju korištenje digitalnih blizanaca poslovnih procesa za praćenje planova razvoja proizvoda. Cilj je omogućiti kontinuirano praćenje procesa razvoja proizvoda u stvarnom vremenu te postići da se predvide bilo kakva odstupanja tijekom izvođenja procesa razvoja proizvoda. Istraživanje provedeno od strane autora pokazuje kako se BPMN može koristiti za izradu detaljnih procesnih modela koje digitalni blizanci repliciraju i prate u stvarnom vremenu. Autori tvrde da ova integracija pruža poboljšanu razinu kontrole nad procesima, omogućujući rano otkrivanje odstupanja i proaktivne intervencije.

Na slici 1 u nastavku se nalazi prikaz osnovnih elemenata notacije za upravljanje poslovnim procesima. Osnovni oblici su tako aktivnosti koje mogu biti obični zadaci ili podprocesi, zatim događaji koji mogu biti početni događaj, međudogađaj, završni događaj i mjerač vremena. Zatim, odluke koje mogu ekskluzivne, inkluzivne ili paralelne, te staze koje mogu biti „bazen“ ili obična staza.



Slika 1 Osnovni oblici u BPMNu (interfacing, ožujak 2025)

Neovisno o sve većoj popularnosti primjene BPMN-a kao osnove za digitalne blizance, nije svaki poslovni proces jednako pogodan za simulacijsko modeliranje i digitalne blizance temeljene na BPMN notaciji. Procesi koji su pogodni za takav pristup karakterizirani su jasno definiranim tokovima aktivnosti, strukturiranom logikom odlučivanja, ograničenim brojem iznimaka te

dostupnošću relevantnih podataka u stvarnom vremenu. Takvi procesi najčešće uključuju ponavljajuće i predvidljive operacije, poput proizvodnih procesa, logističkih lanaca, upravljanja zalihama, planiranja proizvodnje i servisnih intervencija. U tim slučajevima BPMN omogućuje stvaranje standardiziranih modela procesa koji se mogu precizno mapirati u digitalne blizance, čime se osigurava učinkovita simulacija i mogućnost dinamičkog prilagođavanja na temelju stvarnih podataka. S druge strane, procesi koji uključuju visok stupanj nestrukturiranosti, improvizacije ili kompleksnu međuljudsku interakciju, poput procesa strateškog odlučivanja ili inovacijskog dizajna, manje su prikladni za modeliranje pomoću BPMN-a i kasniju implementaciju digitalnih blizanaca. U takvim slučajevima teško je jednoznačno definirati sve tokove, uvjete i ishode, što ograničava mogućnost njihove precizne formalizacije, simulacije i praćenja u stvarnom vremenu. Također, procesi koji ovise o implicitnim znanjima, kontekstualnim procjenama ili koji nemaju dostupne digitalne ulazne podatke, nisu prikladni za izgradnju funkcionalnog digitalnog blizanca.
Iz tog je razloga ključno pri izboru poslovnog procesa za modeliranje i digitalizaciju temeljenu na BPMN-u provesti preliminarnu analizu složenosti i stupnja strukturiranosti kako bi se procijenila izvedivost i korisnost implementacije digitalnog blizanca.

3.3 Integracija proizvodnih digitalnih blizanaca i IoT-a za unapređenje alata za modeliranje poslovnih procesa

Još jedan značajan napredak u ovom području je integracija digitalnih blizanaca s uređajima Interneta stvari za stvaranje povezanih, inteligentnih sustava. Sreedharan i sur. (2025) istražuju kako digitalni blizanci i industrijski IoT (IIoT) mogu revolucionirati rudarsku industriju. Autori u radu predstavljaju prototip platforme za industrijski IoT i digitalnog blizanca za optimizaciju industrijske automatizacije u tradicionalnim rudarskim operacijama. U radu se predlaže referentna arhitektura digitalnog blizanca koja koristi notaciju za modeliranje poslovnih procesa (BPMN) i simulacijske tehnike za provjeru valjanosti te arhitekture u kontekstu rudarstva. Iako se njihovo istraživanje fokusira na fizičke operacije, principi koje opisuju mogu se primijeniti i na BPM. Kombinacijom IoT podataka s digitalnim blizancima organizacije mogu postići praćenje u stvarnom vremenu i prediktivne uvide, povećavajući sposobnost BPM-a za optimizaciju procesa i donošenje odluka.

Ding i sur. (2024) prikazuju kako se koriste blockchain i pametni ugovori za orkestraciju proizvodnih sustava s integriranim digitalnim blizancima. Autori predstavljaju arhitekturu koja se temelji na BPMN-u za integraciju pametnih ugovora i sustava digitalnih blizanaca u proizvodnji. Autori prezentiraju ManuChain4.0, arhitekturu temeljenu na blockchainu 3.0 koja integrira digitalne blizance za proizvodne sustave. Njezina glavna prednost leži u modelu za konfiguraciju resursa i procesa koji osigurava pouzdanost u pregovorima među različitim sudionicima, a sve to omogućuju pametni ugovori u proizvodnji.

U ovom dijelu istaknuta je važnost integracije digitalnih blizanaca s IoT-om i naprednim tehnologijama poput blockchaina i pametnih ugovora kako bi se unaprijedili alati za modeliranje poslovnih procesa. Primjeri iz industrijske prakse, iako primarno usmjereni na fizičke proizvodne sustave, ukazuju na široku primjenjivost principa u kontekstu BPM-a. Korištenjem IoT podataka i digitalnih blizanaca moguće je ostvariti praćenje procesa u stvarnom vremenu, poboljšati analitiku i prediktivno donošenje odluka. Ovi uvidi naglašavaju potencijal tehnološke sinergije za razvoj naprednih, dinamičkih i inteligentnih modela poslovnih procesa.

Primjetan aspekt integracije digitalnih blizanaca s IoT sustavima u industrijskim aplikacijama predstavljaju *edge*[1] (hrv., računarstvo na rubu mreže) i *fog computing*[2] (hrv., računarstvo u oblaku kombinirano s resursima u organizacijama) paradigme koje omogućavaju obradu podataka bliže izvorima generiranja podataka. Prema (Aazam, M. i ostali, 2018), IIoT aplikacije zahtijevaju obradu u stvarnom vremenu, blizinsko skladištenje, ultra-nisku latenciju, pouzdanost i visoku brzinu prijenosa podataka, sve što može biti zadovoljeno *fog computingom*. Istraživanja u ovom području pokazuju da *fog computing* može djelovati kao proširenje centraliziranih sustava u oblaku (eng., *cloud*) prema mrežnom rubu, omogućavajući distribuciju računalnih resursa, usluga i mrežnih kapaciteta bliže IoT uređajima (Bonomi, F. i ostali, 2012). Kako navode (Tao, F. i ostali, 2018), digitalni blizanac je proširena značajka IoT-a koja pruža dublje uvide u bilo koji model za automatizaciju i re-inženjering kako bi se uklonili problemi i učinili ga agilnijim, dok *fog computing*/*edge computing* omogućava korištenje digitalnih blizanaca s poboljšanim performansama. Ova kombinacija tehnologija omogućava stvaranje kognitivnih IoT platformi za industrijske aplikacije koje mogu pružiti računalne resurse bliže korisnicima i uređajima (Verma, S. i ostali, 2017). Za kontekst modeliranja poslovnih procesa, ove paradigme omogućavaju stvaranje hibridnih *cloud-fog* arhitektura koje podržavaju digitalne blizance u stvarnom vremenu, što je ključno za dinamičko upravljanje i optimizaciju poslovnih procesa u industrijskim okruženjima (Liu, Y. i ostali, 2020).

3.4 Upravljanje poslovnim procesima pomoću digitalnih blizanaca unutar CPS arhitekture

Upravljanje poslovnim procesima pomoću digitalnih blizanaca temelji se na integraciji fizičkih entiteta, senzorskih podataka i virtualnog modela procesa unutar kibernetičko-fizičkog sustava (CPS). U takvoj arhitekturi, digitalni blizanac čini kibernetičku komponentu sustava, dok fizički entiteti (strojevi, radne stanice, ljudi) predstavljaju fizičku komponentu. Upravljanje procesima omogućeno je prikupljanjem podataka u stvarnom vremenu iz fizičkog sustava, njihovom analizom u digitalnom blizancu te povratnim utjecajem na tijek stvarnog procesa kroz preporuke, upozorenja ili automatske prilagodbe.

Ključne upravljačke funkcije uključuju nadzor, procjenu performansi, detekciju odstupanja, prediktivnu analitiku i optimizaciju toka procesa. U okviru ovog istraživanja, upravljanje se realizira kroz komunikaciju između stvarnog sustava i digitalnog blizanca proizvodnog poslovnog procesa, pri čemu digitalni blizanac predviđa ponašanje procesa, simulira scenarije i pruža informacije za donošenje odluka. Time se ostvaruje koncept sustava za potporu odlučivanja, gdje čovjek zadržava ulogu nadzora i donošenja ključnih odluka (tzv., eng. *human-in-the-loop*), dok operativne aktivnosti mogu biti djelomično ili potpuno automatizirane.

U istraživanju Talkhestani i sur. (2019) fokusiraju se na razvoj arhitekture inteligentnog digitalnog blizanca unutar kibernetičko-fizičkih proizvodnih sustava. Autori analiziraju postojeće definicije i arhitekture digitalnog blizanca te konceptualiziraju arhitekturu koja pokriva potrebne komponente za realizaciju različitih slučajeva korištenja u inteligentnim automatizacijskim sustavima. Istraživanje naglašava dodanu vrijednost digitalnog blizanca u kontekstu inteligentnih automatizacijskih sustava,

[1] *Edge computing* je paradigma koja omogućava obradu podataka direktno na mrežnom rubu, blizu IoT uređaja i senzora, umjesto slanja podataka u udaljene *cloud* centre.

[2] *Fog computing* je proširenje *cloud computinga* koje distribuira računalne resurse, skladištenje i mrežne usluge između *cloud* centara i *edge* uređaja, stvarajući hijerarhijsku arhitekturu za obradu podataka.

pri čemu se digitalni blizanac ne koristi samo za praćenje i simulaciju, već i za podršku u upravljanju procesima. Predložena arhitektura ima za cilj poboljšanje fleksibilnosti i učinkovitosti proizvodnih procesa kroz implementaciju inteligentnih funkcionalnosti. Agrawal i sur. (2023), predlažu koncept *Razine digitalnog blizanca* (eng. *Levels of Digital Twin*) koji očituje granicu između ljudske kontrole i automatizirane akcije. Oni navode četiri tipa uloga koje digitalni blizanac može igrati (promatrač, analitičar, donositelj odluka i izvršitelj radnje) te podižu razinu automatizacije kroz pet razina: od potpuno ručnih radnji do potpuno autonomnih operacija. Također, van der Aalst (2022) kroz koncept Autonomnog upravljanja izvršenjem procesa (eng. *Autonomous Process Execution Management*) definira šest razina autonomije u upravljanju procesnim izvršenjem, od potpune ručne kontrole do potpune autonomije. U skladu s time, ključna je gradacija razina automatizacije u upravljanju procesima, koju potvrđuju Agrawal i sur. i van der Aalst kroz šest razina autonomnog izvršenja. Prilagodbom tih modela u kontekstu poslovnih procesa dobivamo:

- Na ručnoj razini, operateri vizualiziraju stanje procesa putem digitalnog blizanca te samostalno donose odluke.
- Na poluautomatiziranoj razini, sustav koristi analitiku i predikciju kako bi pružio prijedloge za donošenje odluka.
- Na automatiziranoj razini, odluke se donose autonomno (npr. automatsko preusmjeravanje tokova ako se detektira usko grlo).

U takvom okviru, čovjek ima ključnu ulogu u definiranju pravila, nadzoru odstupanja i reagiranju u nestrukturiranim situacijama, dok digitalni blizanac obavlja u nekim slučajevima složene zadatke, a u drugim rutinske zadatke nadzora, analize i prilagodbe. Znači, uloga čovjeka se ne eliminira, već se redefinira u smjeru nadzora, interpretacije složenijih situacija i optimizacije procesa. Ova distribucija odgovornosti osigurava balans između učinkovitosti automatizacije i fleksibilnosti ljudske prosudbe (*human-in-the-loop* koncept). Time se osigurava neprekidna petlja upravljanja koja povezuje poslovne ciljeve s operativnim izvršenjem procesa.

Uloga digitalnog blizanca ne podrazumijeva zamjenu postojećih sustava za automatsko upravljanje, već njihovu nadogradnju i proširenje funkcionalnosti. Sustavi poput SCADA i dalje zadržavaju ključnu ulogu u upravljanju procesima na operativnoj razini, a to znači u realnom vremenu, neposredno i deterministički. Digitalni blizanci, s druge strane, djeluju na višoj apstraktnoj razini, s fokusom na analitičke i prediktivne zadatke koji uključuju obradu podataka, simulaciju alternativnih scenarija i donošenje složenijih odluka temeljenih na obrascima iz prošlosti i očekivanjima za budućnost. U tom kontekstu, digitalni blizanac ne preuzima upravljačke funkcije SCADA sustava, već ih dopunjuje pružanjem dodatnih informacija i preporuka koje omogućuju optimizaciju proizvodnog sustava, ranije otkrivanje odstupanja te donošenje strateških i taktičkih odluka. Takva podjela odgovornosti omogućuje postizanje balansa između robusnosti tradicionalnih upravljačkih sustava i fleksibilnosti napredne analitike temeljene na digitalnim blizancima.

Predložena metodika za razvoj digitalnih blizanaca u ovom istraživanju podrazumijeva upravo takvu arhitekturu, iako se u istraživanju naglasak stavlja na formalizaciju procesa, prediktivno modeliranje i usporedbu s konvencionalnim simulacijama. U studijama slučaja analiziranima u ovom istraživanju, uloga čovjeka ostaje ključna, primjerice, u situacijama kada proizvodni digitalni blizanac predvidi kvar ili zastoj, konačnu odluku o promjeni stroja, intervenciji ili privremenom zaustavljanju proizvodnog procesa donosi radnik. Na taj način se koncept *human-in-the-loop* operacionalizira kroz praksu, čime se zadržava fleksibilnost i odgovornost ljudskog djelovanja u kritičnim trenucima procesa.

4 Plan istraživanja

4.1 Obuhvat i ograničenja

Ovo istraživanje bavi se razvojem metodike za izradu digitalnih blizanaca proizvodnih poslovnih procesa s naglaskom na integraciju prediktivnog modela održavanja opreme temeljenog na neuronskim mrežama. Cilj metodike je omogućiti modeliranje, analizu i optimizaciju proizvodnih procesa kroz digitalne blizance, čime se poboljšava učinkovitost i smanjuju operativni rizici. Metodika će biti validirana kroz dvije studije slučaja: proizvodnju dronova i proizvodnju mobilnih uređaja. Važno je naglasiti da uspostava studija slučaja zahtijeva detaljnu suradnju s industrijskim partnerima, pristup realnim podacima, kao i dubinsko razumijevanje konkretnih proizvodnih procesa, što čini taj postupak složenim. Svaka studija slučaja uključuje prikupljanje i obradu podataka te iterativno testiranje i validaciju rješenja. Zbog toga je broj studija slučaja u ovom istraživanju ograničen na dvije, s obzirom na to da svaka od njih zahtijeva značajan vremenski i organizacijski angažman, kao i tehničku kompleksnost provedbe. Predloženi pristup uključuje modeliranje poslovnih procesa, integraciju podataka iz proizvodnog okruženja te primjenu neuronskih mreža za predviđanje kvarova i optimizaciju održavanja opreme. Naglasak se stavlja na prilagodljivost metodike različitim industrijama, kao i na mogućnost njenog proširenja kroz daljnji razvoj digitalnih blizanaca u kontekstu prediktivne analitike i autonomnog odlučivanja.

Unatoč širokom obuhvatu, metodika ima određena ograničenja. Prvo, za treniranje neuronske mreže će se koristiti sintetički podaci iz otvorenog skupa podataka jer je pravu količinu informacija, nužnu za kvalitetno treniranje modela, teško prikupiti. Podaci sa senzora u stvarnim proizvodnim sustavima često su nedostupni zbog tehničkih, sigurnosnih ili vlasničkih ograničenja, što može otežati razvoj prediktivnih modela temeljenih na stvarnim podacima iz industrijskih okruženja. Nadalje, implementacija digitalnih blizanaca zahtijeva složenu integraciju s postojećim poslovnim sustavima i senzorskim mrežama, što može predstavljati izazov. Također, iako metodika pokriva industriju proizvodnih procesa, njezina primjenjivost na druge sektore može zahtijevati dodatne prilagodbe ovisno o specifičnostima pojedinih poslovnih sustava.

4.2 Ciljevi i hipoteza

S obzirom na to da je kroz pregled literature utvrđeno kako ne postoji definiran način na koji se razvijaju proizvodni digitalni blizanci, ciljevi ovog istraživanja su:

1. Oblikovati metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa proširenjem postojeće metodike modeliranja poslovnih procesa.
2. Razviti digitalnog blizanca proizvodnog poslovnog procesa primjenom metodike za oblikovanje digitalnih blizanaca proizvodnih poslovnih procesa.

Znači, nakon što će se oblikovati metodika za razvoj proizvodnih digitalnih blizanaca, temeljem nje će se razviti proizvodni digitalni blizanci dvaju slučajeva korištenja.

Istraživačko pitanje ovog istraživanja glasi:

Kako dizajnirati metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa?

Postojeće metodike za upravljanje poslovnim procesima će se proširiti kako bi se dizajnirala metodika. Znači, temeljem postojećih metodika će se razviti nova metodika za razvoj proizvodnih digitalnih blizanaca.

Hipoteza istraživanja u ovom istraživanju glasi: *Digitalni blizanac proizvodnog poslovnog procesa razvijen proširenom metodikom modeliranja poslovnih procesa emulira realni proizvodni poslovni proces kako se on odvija u stvarnosti.* Istraživačka hipoteza bit će potvrđena ili opovrgnuta primjenom metoda studije slučaja i simulacijskim eksperimentom.

4.3 Struktura rada

U prvom poglavlju, *Uvod*, opisuje se područje istraživanja, definiraju se problem i istraživački izazov kojima se rad bavi. Poglavlje *Upravljanje poslovnim procesima i alati za modeliranje poslovnih procesa* daje pregled osnovnih koncepata upravljanja poslovnim procesima i modeliranja procesa. U njemu se iznosi pregled postojećih istraživanja iz područja upravljanja poslovnim procesima i alata za modeliranje poslovnih procesa. Sljedeće poglavlje, *BPM, BPMN i digitalni blizanci*, objašnjava koncept upravljanja poslovnim procesima (BPM) i modeliranja pomoću notacije BPM-a te uvodi pojam digitalnih blizanaca u kontekst poslovnih procesa. Potpoglavlje *Upravljanje poslovnim procesima i digitalni blizanci poslovnih procesa* detaljnije razmatra povezanost digitalnih blizanaca s upravljanjem poslovnim procesima i njihovom primjenom u industrijskom okruženju.
U potpoglavlju *BPMN kao osnova za digitalne blizance proizvodnih poslovnih procesa* analizira se upotreba BPMN-a kao načina modeliranja za izradu spomenutih. Potpoglavlje *Integracija proizvodnih digitalnih blizanaca i IoT-a za unapređenje alata za modeliranje poslovnih procesa* opisuje kako se Internet stvari (IoT) mogu koristiti za poboljšanje rada digitalnih blizanaca i njihovu povezanost s fizičkim sustavima. Poglavlje *Plan istraživanja* opisuje metodološki okvir rada, dok se u potpoglavljima *Obuhvat i ograničenja* te *Ciljevi i hipoteza* definiraju istraživački ciljevi, istraživačko pitanje i hipoteza istraživanja. U poglavlju *Istraživačka metodologija* opisuje se metodologija istraživanja korištena u provedbi istraživanja. Zatim slijedi poglavlje *Metoda simulacijskog eksperimenta*, gdje se razmatra simulacijski pristup u modeliranju poslovnih procesa i evaluaciji digitalnih blizanaca. Poglavlje *Proširenje standardnog pristupa modeliranja poslovnih procesa za digitalne blizance* iznosi zašto dolazi do potrebe proširenja postojećeg standarda modeliranja poslovnih procesa u kontekstu razvoja digitalnih blizanaca. Sljedeći dio rada usmjeren je na prikaz praktičnih primjena kroz studije slučaja. Poglavlje *Studija slučaja: Inovacijski park Biel/Bienne* opisuje poslovni proces proizvodnje dronova, dok *Studija slučaja: Fakultet strojarstva i brodogradnje (FSB) Sveučilišta u Zagrebu* opisuje proces proizvodnje mobilnih uređaja. Poglavlje *Koraci razvoja digitalnog blizanca proizvodnog poslovnog procesa* opisuje faze razvoja digitalnog blizanca od analize do optimizacije. U poglavlju *Primjena umjetne inteligencije u razvoju digitalnih blizanaca* objašnjava se kako se umjetna inteligencija koristi za prediktivno održavanje unutar digitalnih blizanaca. Prikazuju se metode prikupljanja i obrade podataka te razvoj prediktivnih modela. Poglavlje *Metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa* predstavlja „vodič“ za izgradnju digitalnih blizanaca temeljem koje su i napravljeni proizvodni digitalni blizanci čiji su rezultati prikazani u nastavku. Zatim se radi usporedba simuliranih modela s proizvodnim digitalnim blizancima. U poglavlju *Unaprjeđenje postojećih alata za upravljanje poslovnim procesima* analizira se mogućnost poboljšanja postojećih alata za modeliranje poslovnih procesa. Poglavlje *Primjenjivost predložene metodike razvoja proizvodnih digitalnih blizanaca* prikazuje važnost i korist predložene metodike. U poglavlju *Znanstveni doprinos istraživanja* istaknuti su ključni znanstveni doprinosi proizašli iz istraživanja, a posljednje poglavlje, *Zaključak*, donosi sažetak istraživanja, glavne rezultate i smjernice za buduća istraživanja. Na kraju rada nalaze se popisi slika i tablica, korištena literatura te prilozi s korištenim kodom.

5 Istraživačka metodologija

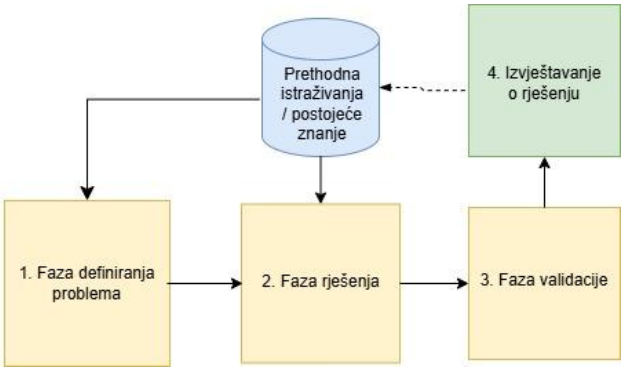
Ovaj doktorat temelji se na paradigmi znanstvenog istraživanja dizajna (eng., ***D**esign **S**cience **R**esearch*, DSR) prema (Hevner, A.R. i ostali, 2004) i (Kuechler, B. & Vaishnavi, V., 2008), koji je prikladan za istraživanja usmjerena na razvoj inovativnih rješenja u složenim poslovnim i tehnološkim okruženjima. U području znanstvenog istraživanja dizajna najčešće se referiraju pristupi autora kao što su Alan Hevner, Jan vom Brocke (vom Brocke, J. i ostali, 2020), Vijay Vaishnavi i Kuechler (Vaishnavi, V. & Kuechler, B., 2004) te Roel Wieringa (Wieringa, R.J., 2014). Hevner, A.R. i ostali (2004) postavili su temeljni okvir DSR-a koji se temelji na razvoju i evaluaciji istraživačkih artefakata unutar sustava informacija. DSR metodologija primijenjena je zbog njenog fokusa na izgradnju i evaluaciju artefakata, pri čemu je u ovom istraživanju artefakt metodika za razvoj digitalnog blizanca proizvodnog poslovnog procesa s integriranim modelom prediktivnog održavanja. Ova metodologija omogućuje iterativni razvoj i eksperimentalnu provjeru modela, čime se osigurava njegova primjenjivost u realnim proizvodnim sustavima.

Primjena DSR-a u ovom istraživanju opravdana je time što omogućuje strukturirani pristup dizajnu, implementaciji i validaciji metodike, osiguravajući da razvijeni model ne bude samo teorijski koncept, već praktično primjenjivo rješenje. Također, DSR metodologija naglašava povratnu vezu između dizajna i evaluacije, čime se postiže optimizacija modela proizvodnog digitalnog blizanca na temelju empirijskih rezultata. Na taj način, istraživanje doprinosi i teorijskom razvoju koncepta digitalnih blizanaca i njihovoj praktičnoj implementaciji u kontekstu industrijske proizvodnje.

Istraživačka metodologija ovog rada temelji se na kombinaciji dobrih praksi za provođenje istraživanja u području softverskog inženjerstva koje je predložila Mary Shaw (Shaw, M., 2003), slično kao što je to učinio Ivan Švogor (Švogor, Ivan, 2016) u svojoj doktorskoj disertaciji. U ovom istraživanju odabran je pristup Švogora jer je tematski i metodološki najbliži kontekstu istraživanja provedenog u ovom doktoratu. Švogor je razvijao i evaluirao artefakte unutar tehničkih i inženjerskih sustava. S obzirom na to da se u ovom doktoratu razvija digitalni blizanac proizvodnog poslovnog procesa koji uključuje tehnički sustav, pristup kolege Švogora pokazao se prikladnim jer omogućuje strukturiranje istraživanja, razvoj artefakta i njegovu validaciju u realnom, tehnički utemeljenom okruženju. Dodatno, Švogorov pristup osigurava metodološku konzistentnost s temeljnim DSR načelima, ali uz zadržavanje praktične primjenjivosti u tehničkim sustavima, što je bilo ključno za ostvarenje ciljeva istraživanja. Švogor navodi tri prepoznatljive i ponovljive faze:

1. Faza definiranja problema: U ovoj se fazi identificira i formulira istraživački problem na temelju postojećih znanstvenih spoznaja. Istraživački problem koji se rješava ovom doktorskom disertacijom je da postojeće metodike za upravljanje poslovnim procesima nisu prilagođene razvoju digitalnih blizanaca proizvodnih poslovnih procesa te da ne postoji metodika koja objašnjava kako razviti digitalnog blizanca proizvodnog poslovnog procesa.
2. Faza rješenja: Tijekom ove faze predlaže se rješenje prethodno definiranih istraživačkih pitanja koristeći postojeće spoznaje i poznate principe. Ovaj rad predlaže metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa i njezine korake temeljene na postojećim metodikama.

3. Faza verifikacije: U ovoj fazi simulacijskim eksperimentom se provjerava da li digitalni blizanac proizvodnog poslovnog procesa realno emulira proizvodni poslovni proces kako se on odvija u stvarnosti, pri čemu "emulira" znači da se model digitalnog blizanca proizvodnog poslovnog procesa u svim relevantnim aspektima ponaša kao stvarni proizvodni poslovni proces. Svrha simulacijskog eksperimenta je dobiti uvid u funkcioniranje proizvodnog poslovnog procesa i predvidjeti njegovo ponašanje u budućnosti te donijeti operativne odluke prije nego što se dogode nepredvidljive okolnosti. U sklopu ove faze razvija se proizvodni digitalni blizanac koji prati stanje stvarnog poslovnog procesa te se radi simulirani model proizvodnog poslovnog procesa temeljem povijesnih podataka.

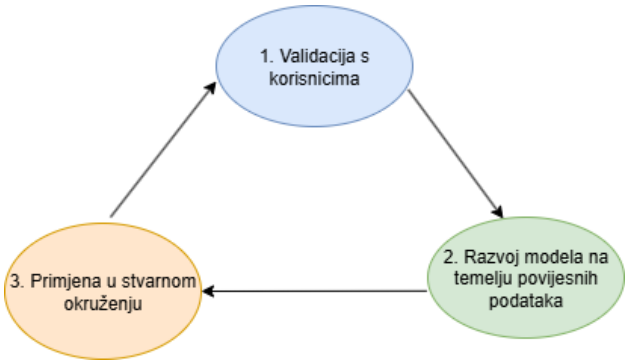


Slika 2 Trofazna istraživačka metodologija korištena od strane Švogora (2016)

Osim nabrojanih faza, prema slici 2 može se primijetiti i korak *Izvještavanje o rješenju* gdje se rezultati istraživanja objavljuju u obliku znanstvenih radova. Pošto je istraživanje cirkularni proces koji se uvijek nadograđuje novim saznanjima, sva prethodno prikupljena istraživanja i postojeće znanje se koriste kako bi se problem bolje definirao ili potvrdio te sukladno njemu, ponudilo i nadogradilo rješenje za taj problem koje se zatim validira.
Faza izvještavanja o rješenju predstavlja važnu ulogu u ovom procesu jer osigurava učinkovitu komunikaciju rezultata širem stručnom krugu. Prema Mary Shaw (2003) jasno i detaljno izvještavanje povezuje istraživanje s praksom. Ono omogućuje istraživačima da precizno objasne svoje metodike, opravdaju pristupe i istaknu važnost svojih doprinosa. Ova faza ne samo da potvrđuje predloženo rješenje već i potiče evaluaciju od strane kolega, reprodukciju simulacijskog eksperimenata i poboljšanje rezultata, čime se stvara poticajno akademsko okruženje koje koristi cijelom području. Korištenje postojećeg znanja i prethodnih istraživanja je važno jer pruža temelje za razvoj i validaciju novih ideja. Shaw naglašava potrebu za kontekstualizacijom istraživanja u odnosu na postojeća saznanja kako bi se identificirali nedostaci, izazovi i mogućnosti za unaprjeđenje. Sintetiziranjem prethodnih rješenja istraživači mogu izbjeći nepotrebno ponavljanje, nadograditi se na provjerene metodike i sustavno unaprijediti stanje znanosti. Ova praksa također povećava vjerodostojnost i relevantnost istraživanja jer pokazuje razumijevanje povijesnog konteksta i trenutnih trendova u području.

Navedene faze razvoja prema Švogoru odgovaraju tipičnom pristupu istraživanju unutar tehničkih i informacijskih znanosti, pri čemu se najprije definira problem, zatim predlaže rješenje, a potom provodi evaluacija predloženog rješenja. Slijedom te logike, i ovo istraživanje može biti strukturirano kroz tri faze koje prate iterativni pristup:

1. Validacija s korisnicima – faza u kojoj se definiraju istraživačke aktivnosti. U ovoj fazi definiraju se aktivnosti proizvodnih poslovnih procesa, koje radnici uključeni u studije slučaja pregledavaju i potvrđuju da vjerno prikazuju stvarni način izvođenja procesa.
2. Razvoj modela na temelju povijesnih podataka – faza u kojoj se izrađuje funkcionalni digitalni blizanac proizvodnog poslovnog procesa te trenira prediktivni model za potrebe optimizacije.
3. Primjena u stvarnom okruženju – faza u kojoj stvarni korisnici (radnici) testiraju predloženi model u praksi, a povratne informacije koje daju služe za procjenu rješenja.



Slika 3 Faze istraživanja koje prate iterativni pristup

Slika 3 ilustrira iterativni pristup istraživanju kroz tri povezane faze: validaciju s korisnicima, razvoj modela na temelju povijesnih podataka i primjenu u stvarnom okruženju. Strelice između faza ukazuju na kružni tok aktivnosti, čime se naglašava važnost povratne veze u smislu da se rezultati i povratne informacije iz stvarnog okruženja vraćaju u fazu validacije, gdje se mogu dodatno usavršiti modeli i planirane aktivnosti. Na taj način, istraživanje se ne provodi linearno, već kao dinamičan ciklus stalnog unaprjeđenja kroz učenje, evaluaciju i prilagodbu.

Za izradu simuliranog modela proizvodnog poslovnog procesa koristit će se WebSphere Business Modeler verzije 7.0. WebSphere Business Modeler omogućava modeliranje, simulaciju i analizu poslovnih procesa te izradu izvještaja o poslovnim procesima kako bi pomogao u optimizaciji izvedbe poslovnih procesa (IBM, 2025a). Osim za izradu simuliranog modela, navedeni alat se koristi i za provođenje dinamičke analize rezultata simulacije kako bi se utvrdila uspješnost izvedbe simulacije te da li simulirani model odgovara izvođenju stvarnog proizvodnog poslovnog procesa. Na tržištu postoji više alata za modeliranje i simulaciju poslovnih procesa, koji se razlikuju prema funkcionalnostima, cijeni i razini podrške za dinamičku simulaciju. Među najpoznatijim komercijalnim alatima ističu se Bizagi Modeler, ARIS Architect[3], SAP Signavio Process Manager[4] i Simul8[5]. Od navedenih, Bizagi Modeler omogućuje intuitivno modeliranje procesa prema BPMN standardu, no njegova osnovna verzija ne podržava izvođenje simulacija, već služi pretežito za dokumentiranje i analizu procesa. ARIS i SAP Signavio nude mogućnosti napredne analize i vizualizacije, uključujući djelomičnu podršku za simulaciju, no puni opseg simulacijskih funkcionalnosti dostupan je isključivo u skupljim licenciranim paketima. Simul8, iako prvenstveno orijentiran na opće poslovne i operativne simulacije, također omogućuje izradu simulacijskih modela, ali nije posebno optimiziran za poslovne procese temeljene na BPMN notaciji. U kontekstu ovog istraživanja, IBM WebSphere Business Modeler odabran je zbog integriranih mogućnosti

[3] Dostupno na: https://ariscommunity.com/university/aris-architect.
[4] Dostupno na: https://www.signavio.com/products/process-manager/ .
[5] Dostupno na: https://www.simul8.com/.

modeliranja, izvođenja simulacija i analize performansi poslovnih procesa u istom razvojnom okruženju. Dodatno, alat podržava izvođenje dinamičkih analiza koje su važne za usporedbu rezultata simulacije s rezultatima digitalnog blizanca.

U ovom istraživanju za provjeru hipoteze su upotrjebljena dva slučaja korištenja (eng. *use cases*) koja omogućuju razumijevanje kompleksnijih sustava u njihovom stvarnom kontekstu kao što su to učinili autori (Marmolejo-Saucedo, J.A., 2020), (Torres, J. i ostali, 2024), (Cao, X. i ostali, 2024), (Yan, J. i ostali, 2024). S obzirom da je cilj predložiti metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa, odabrana su dva reprezentativna slučaja iz proizvodnog sektora: proizvodnja dronova u Inovacijskom parku i simulirana proizvodnja mobilnih uređaja u "Pametnoj tvornici", koji će biti detaljnije opisani u nastavku ovog rada. Prvi slučaj predstavlja proizvodni proces koji uključuje napredne proizvodne tehnologije poput 3D printanja, robotike i RFID sustava, dok drugi slučaj prikazuje školsku proizvodnu liniju koja simulira industrijski proces proizvodnje mobilnih uređaja. Ova dva slučaja odabrana su zato što predstavljaju različite razine složenosti i tehnološke integracije u proizvodnim procesima, što omogućuje sveobuhvatnu analizu primjene digitalnih blizanaca u pametnoj industriji.
Za izradu digitalnih blizanaca proizvodnih poslovnih procesa korišten je alat Camunda Platform (*Camunda: The Universal Process Orchestrator*, 2024). Camunda je platforma otvorenog koda za modeliranje i automatizaciju poslovnih procesa koja se temelji na BPMN 2.0 (eng., *Business Process Model and Notation*) standardu. Platforma omogućuje modeliranje, izvršavanje i nadzor poslovnih procesa u stvarnom vremenu, što ju čini pogodnom za razvoj digitalnih blizanaca. Digitalni blizanci su implementirani kroz kombinaciju BPMN modela u Camundi i Python skripti koje su omogućile automatizaciju određenih aktivnosti u procesu. Python skripte korištene su za obradu podataka i aktiviranje određenih akcija u proizvodnim procesima. Ova kombinacija alata omogućila je stvaranje vjerodostojne digitalne reprezentacije fizičkih proizvodnih procesa, gdje Camunda služi za upravljanje tijekom procesa i poslovnom logikom, dok Python skripte osiguravaju dodatnu funkcionalnost kroz automatizaciju i obradu podataka.

Na tržištu postoji niz alata za modeliranje i upravljanje poslovnim procesima, uključujući otvorena (eng., *open source*) i komercijalna rješenja. U tablici 1 se nalazi prikaz usporedbe alata za modeliranje i upravljanje poslovnim procesima. Među alternativama Camundi često se ističu alati kao što su Bonita BPM, jBPM, Activiti i Flowable (svi otvoreni), te komercijalni alati poput IBM Business Automation Workflow, Appian i Bizagi.

Alat	**Vrsta licence**	**Podrška za BPMN 2.0**	**Fleksibilnost[6]**	**Integracija s Pythonom[7]**	**Pogodno za istraživačke prototipove[8]**
Camunda	*Open source* (i komerc.)	Da	Visoka	Izvrsna (REST API + skripte)	Da

[6] Odnosi se na mogućnost prilagodbe i proširivanja funkcionalnosti alata.
[7] Vrednovana je prema mogućnosti poziva Python skripti unutar tijeka procesa (nativno ili putem REST API-ja).
[8] Komercijalni alati nisu optimalni za istraživanje zbog licenciranja.

Flowable	*Open source*	Da	Visoka	Dobra (REST API + konfiguracija)	Da
jBPM	*Open source*	Da	Visoka	Dobra (REST API + skripte)	Da
Bonita BPM	*Open source* (i komerc.)	Da	Umjerena	Dobra (REST API + konektori)	Da
Activiti	*Open source*	Da	Umjerena	Dobra (REST API + biblioteke)	Da
IBM Business Automation Workflow	Komercijalna	Da	Visoka (zatvoreni sustav)	Dobra (REST API + *middleware*)	Ne
Appian	Komercijalna	Da	Visoka (zatvoreni sustav)	Dobra (REST API + integracije)	Ne
Bizagi	Komercijalna	Da	Visoka (zatvoreni sustav)	Dobra (REST API + konektori)	Ne

Tablica 1 Usporedba alata za modeliranje i upravljanje poslovnim procesima

Bonita BPM, jBPM i Flowable, primjerice, nude podršku za BPMN 2.0 standard te omogućuju proširivost i integraciju s vanjskim komponentama. Flowable, koji je nastao kao nastavak originalnog Activiti projekta, odlikuje se aktivnom zajednicom i modernijim arhitektonskim pristupom. Kao i ostali BPM alati, omogućuje integraciju s vanjskim sustavima uključujući Python komponente kroz REST API-je. Bonita BPM koristi "*zero-coding approach*" s konceptom konektora koji može olakšati integraciju s vanjskim sustavima, dok jBPM omogućuje fleksibilnu integraciju kroz različite pristupe uključujući skriptiranje i vanjske servise. Komercijalni alati poput IBM-ova rješenja, Appiana i Bizagija nude bogata vizualna sučelja i napredne analitičke funkcionalnosti. Iako mogu pružiti značajne mogućnosti prilagodbe, često zahtijevaju licenciranje i mogu biti manje prikladni za istraživačke prototipove koji zahtijevaju potpunu kontrolu nad procesnom logikom i integracijom vanjskih modula bez dodatnih troškova.

Zbog otvorenog koda, modularne arhitekture i snažne podrške za BPMN standard, Camunda (*Community Edition*) se pokazala kao optimalno rješenje za potrebe ovog istraživanja. Camunda *Community Edition* je otvoreno rješenje dostupno pod Apache 2.0 i MIT licencama, dok postoji i komercijalna verzija s dodatnim funkcionalnostima za *Enterprise* potrebe. Za potrebe ovog istraživanja odabrana je navedena verzija Camunde jer omogućava integraciju s Python skriptama i

povezivanje s drugim sustavima, što je ključno za izgradnju funkcionalnog digitalnog blizanca proizvodnog poslovnog procesa.

Za potrebe provedbe istraživanja prethodnih dostignuća u području digitalnih blizanaca i prediktivnog održavanja, pretraživane su relevantne znanstvene baze podataka, uključujući *Web of Science* (WoS), Scopus i IEEE Xplore. Cilj pregleda postojećih istraživanja bio je identificirati postojeće pristupe modeliranju digitalnih blizanaca poslovnih procesa, metode prediktivnog održavanja te primjenu neuronskih mreža u analizi kvarova industrijskih strojeva. Pregled literature omogućio je uvid u trenutne znanstvene trendove i izazove, što je poslužilo kao temelj za razvoj metodološkog okvira istraživanja. Rezultati prethodnih istraživanja nalaze se u poglavljima: „Upravljanje poslovnim procesima i alati za modeliranje poslovnih procesa“, „BPM, BPMN i digitalni blizanci“, „Metoda simulacijskog eksperimenta“, „Općenito o digitalnim blizancima“; „Proširenje standardnog pristupa modeliranja poslovnih procesa za digitalne blizance“, „Primjena umjetne inteligencije u razvoju digitalnih blizanaca“ i „Metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa“.

6 Metoda simulacijskog eksperimenta

6.1 Općenito o simulacijskom eksperimentu

Simulacijski eksperimenti se koriste u različitim disciplinama, a pružaju platformu za istraživanje, optimizaciju i predviđanje ponašanja složenih sustava koje bi inače bilo teže istražiti u stvarnom svijetu. Ovakvi eksperimenti nude način navigacije u scenarijima "što ako", potvrđuju teorije i usmjeravaju procese donošenja odluka (Baeldung, 2024). U simulacijskim eksperimentima su ključni modeli koji nude različite perspektive i mogućnosti za razumijevanje i analizu složenih sustava.
Simulacijski eksperiment u ovom doktoratu se koristi kod identifikacije da li simulirani model poslovnog procesa oponaša poslovni proces u stvarnom svijetu i da li digitalni blizanac proizvodnog poslovnog procesa oponaša stvarni poslovni proces u realnom vremenu. Svrha simulacijskog eksperimenta je dobiti uvid u funkcioniranje proizvodnog poslovnog procesa i predvidjeti njegovo ponašanje u različitim uvjetima te donijeti operativne odluke prije nego što se dogode nepredvidljive okolnosti.
Primjena simulacijskog eksperimenta pomaže riješiti barijeru koju primjena „pravog“ eksperimenta ne može, poput provedbe eksperimentalnih operacija prometnih nesreća pri velikim brzinama. Simulacijski eksperiment se može provesti bilo kada i bilo gdje, što je pogodno kako bi poboljšala učinkovitost, sigurnost i točnost inženjerskog pristupa te ušteda resursa (Wu, Q. i ostali, 2022).
Tijekom provedbe simulacijskog eksperimenta potrebno je u obzir uzeti sljedeće:

- Smanjiti količinu generiranih podataka uz nastojanje da se zadrži potrebna količina informacija bez gubitka točnosti informacija potrebnih za optimizaciju simulacije;
- Osigurati visoku brzinu pretraživanja tih informacija;
- Upotrijebiti paralelizaciju tijekom simulacijskog eksperimenta nad, npr., digitalnim blizancem;
- Smanjiti suvišne podatke;
- Spriječiti mogući gubitak podataka tijekom provedbe simulacijskog eksperimenta (Raska, P. i ostali, 2021).

Načini kako smanjiti količinu generiranih podataka tijekom optimizacije simulacije, a bez gubitka važnih informacija su kombiniranje različitih pristupa globalne optimizacije, programiranje i tehnike „velikih podataka“ (eng., „*big data techniques*“).
Simulacijski eksperimenti se mogu podijeliti u sljedeće kategorije (Raska, P. i ostali, 2021):

1. Simulacijski eksperiment. Odnosi se na izvođenje modela simulacije diskretnog događaja i izračunavanje vrijednosti funkcije cilja korištenjem izlaza (eng., „*outputs*“) iz modela simulacijskog modela.
2. Optimizacijski eksperiment. Provodi se s određenom postavkom metode optimizacije kako bi se pronašao optimum funkcije cilja (npr., kako bi se pronašlo najbolje moguće rješenje modeliranog problema pomoću digitalnog blizanaca).
3. Serije. Ponavljanje optimizacijskih eksperimenata s određenom optimizacijom postavljanja metode (kako bi se testiralo ponašanje optimizacijske metode i djelomičnog smanjenja pogrešnih postavki parametara optimizacijske metode). Testiraju se različite postavke optimizacijskih metoda te se repliciraju optimizacijski eksperimenti s konkretnim postavkama

optimizacijske metode kako bi se smanjio utjecaj slučajne implementacije na optimizacijsku metodu (Raska, P. i ostali, 2021).

Dizajniranje simulacijskog eksperimenta je sustavni proces koji kreće od artikuliranja svrhe eksperimenta do tumačenja rezultata. Faze dizajniranja simulacijskog eksperimenta su (Baeldung, 2024):

1. Faza modeliranja u kojoj se odabire i izrađuje model temeljem prikupljenih podataka o entitetu kojeg se modelira. Zatim se isti model validira i verificira na način da se uspoređuje idejno rješenje (*output*) s prvotno prikupljenim podacima.
2. Eksperimentalna faza u kojoj se odabiru odgovarajući eksperimentalni uvjeti te se eksperiment ponavlja (simulacija se može pokretati više puta s istim inicijalnim uvjetima i vrijednostima parametara).
3. Faza analize podataka u kojoj se analiziraju podaci te se dokumentiraju postignuti rezultati. Tu se gleda sažetak statistike te se dokumentiraju metode i rezultati i izvještava se o rezultatima eksperimenta.

Koraci simulacijskog eksperimenta su tako (Baeldung, 2024):

1. Odabir odgovarajućeg modela. Odabir prikladnog modela za simulacijski eksperiment je važna odluka koja značajno utječe na kvalitetu rezultata i njihovu korist. Podkoraci su:
    a. Razumijevanje sustava
    b. Definiranje ciljeva
    c. Razmatranje dostupnih podataka
    d. Procjena kompromisa
    e. Iterativna dorada modela
2. Verifikacija i validacija modela. Nakon što je model odabran i izrađen, potrebno je provjeriti njegovu točnost i pouzdanost. To se radi putem verifikacije i validacije modela. Kako usavršavamo i poboljšavamo model te kako novi podaci postaju dostupni, potrebno je ponavljati ove korake kako bismo osigurali da model ostane točan i pouzdan. Važno je za napomenuti kako je dio procesa validacije razumijevanje i uvažavanje ograničenja i pretpostavki modela. Podkoraci su:
    a. Verifikacija modela
    b. Validacija modela
3. Eksperimentalni dizajn. Nakon verifikacije i validacije modela, sljedeći korak u simulacijskom eksperimentu je dizajn stvarnog eksperimenta. To uključuje određivanje načina na koji će se rukovati varijablama i parametrima modela te kako će se mjeriti i analizirati rezultati. Cilj eksperimentalnog dizajna bi trebao biti maksimizacija količine i kvalitete informacija koje se dobivaju iz simulacije uz minimizaciju potrebnih resursa. Podkoraci su:
    a. Postavljanje početnih uvjeta
    b. Odabir vrijednosti parametara
    c. Odlučivanje o replikacijama
    d. Odabir izlaznih vrijednosti
    e. Analiza rezultata
4. Analiza podataka. Analiza podataka se odnosi na interpretaciju rezultata, razumijevanje njihovih implikacija te učinkovitom prenošenju tih informacija. Podaci predstavljaju ponašanje sustava pod specifičnim uvjetima. Podkoraci su:
    a. Početni pregled podataka

   b. Statistička analiza podataka
   c. Interpretacija rezultata
   d. Analiza osjetljivosti
   e. Izvještavanje o rezultatima
5. Izrada dokumentacije. Ispravna dokumentacija je ključan aspekt dizajniranja simulacijskog eksperimenta. Ona pomaže osigurati ponavljanje istraživanja, olakšava suradnju s drugima te omogućava buduće reference i poboljšanja. Temeljitim i sustavnim dokumentiranjem simulacijskog eksperimenta povećavamo transparentnost, ponovljivost i vjerodostojnost rada.

6.2 Simulacijski eksperiment i digitalni blizanci

S brzim napretkom informacijsko-komunikacijske tehnologije, virtualni simulacijski eksperimenti temeljeni na digitalnim blizancima su dostignuli određenu razinu zrelosti (Yan, J. i ostali, 2024). Simulacijskim eksperimentom se može potvrditi da digitalni blizanac može odražavati ponašanje fizičkog entiteta i predvidjeti kašnjenje u dinamičkim okruženjima (Lu, Y. i ostali, 2023). Simulacijski eksperiment nad digitalnim blizancem se može koristiti kako bi se iterativno poboljšao, npr., logistički proces čime se poboljšava cjelokupni proizvodni proces kombinacijom virtualnih i stvarnih metoda (Yan, J. i ostali, 2024).

Autori (Lu, Y. i ostali, 2023) predstavljaju okvir temeljen na digitalnom blizancu u kojem se tehnologija digitalnih blizanaca uvodi radi smanjenja troškova upravljanja i optimizacije performansi satelitskih mreža. Istražen je i uspostavljen virtualni model metode raspoređivanja u satelitskim sustavima za simulaciju i predviđanje rezultata kašnjenja pri ponovnom prijenosu. Simulacijski eksperimenti pokazali su da digitalni blizanac može odražavati ponašanje raspoređivanja i predvidjeti kašnjenje u dinamičnim okruženjima.

Simulacijski sustav digitalnog blizanca izgrađen je i korišten u simulacijskim eksperimentima nesreća visokobrzinskih motora, koristeći 3dsMax, Unity3D i C# u članku (Wu, Q. i ostali, 2022). U radu je uspostavljen opći, točan i vremenski osjetljiv model digitalnog blizanca motora za analizu fiksnog, neovisno pobuđenog istosmjernog motora na laboratorijskom stolu, a provedena je eksperimentalna simulacijska primjena kako bi se precizno prikazali parametri eksperimentalnih podataka i stanje motora fiksnog istosmjernog motora u stvarnom vremenu.

Virtualna simulacijska platforma za eksperimentalnu nastavu opisana u radu (Yan, J. i ostali, 2024), razvijena posebno za optimizaciju planiranja u radionicama s montažnim linijama, predstavlja praktični primjer integracije tehnologije digitalnih blizanaca u eksperimentalnu nastavu temeljenu na radionicama te ima važnu ulogu u unaprjeđenju teorijskog znanja studenata i njihovih vještina praktične inovacije. Glavni cilj simulacijskog eksperimenta bio je omogućiti studentima virtualno eksperimentalno okruženje u kojem mogu provjeravati različite strategije raspoređivanja u radionicama s montažnim linijama i usavršavati algoritam raspoređivanja besprijekornom integracijom virtualnih i stvarnih metoda.

U radu (Cao, X. i ostali, 2024) predložena je eksperimentalna metoda modeliranja digitalnog blizanca za montažnu liniju prednjeg i središnjeg kućišta prijenosa. U istraživanju je osmišljena arhitektura sustava temeljenog na digitalnom blizancu, a model digitalnog blizanca konstruiran je izradom

vizualizacijskog modela, logičkog modela i podatkovnog modela montažne linije. Na kraju je proveden simulacijski eksperiment u virtualnom prostoru radi analize postojećih problema u trenutačnoj montažnoj liniji.

U radu (Jinzhi, L. i ostali, 2022) se istražuje koncept kognitivnog digitalnog blizanca (CDT) i njegove temeljne elemente primjenom inženjerskog pristupa sustavima. Kako bi se podržao razvoj kognitivnog digitalnog blizanca, osmišljena je konceptualna arhitektura u skladu sa standardom ISO 42010. Nadalje, pružen je okvir primjene koji koristi graf znanja za usmjeravanje primjena CDT-a. Uz to, predložen je lanac alata kompatibilan s okvirom kako bi se olakšala implementacija CDT-a. Na kraju su autori proveli studiju slučaja temeljenu na simulacijskim eksperimentima kao dokaz koncepta.

Autor članka (Jiang, Y., 2021) kombinira tehnologiju digitalnih blizanaca kako bi konstruirao cjelokupnu strukturu sustava upravljanja i održavanja u građevinskoj industriji, kontroliranog digitalnim blizancem. Osim toga, provodi se inteligentna simulacija građevinskog procesa, a tehnologija digitalnih blizanaca primjenjuje se na upravljanje izgradnjom inteligentnih zgrada. Kroz istraživanje simulacijskih eksperimenata autor je zaključio da model upravljanja izgradnjom inteligentnih zgrada temeljen na digitalnim blizancima, predložen u radu, može na više načina podržati implementaciju inteligentnog građevinskog procesa.

U radu (Wu, Y. i ostali, 2023) predložena je robusna metoda optimizacije u stvarnom vremenu temeljena na digitalnim blizancima u lancu opskrbe. Na temelju kvantitativnog logičkog odnosa između proizvodnje i prodaje proizvoda iz pojedinačnih lanaca, konstruirana je jednadžba stanja sustava lanca opskrbe s logičkim, strukturnim i kvantitativnim karakteristikama te su izgrađeni digitalni blizanci lanaca opskrbe. Autori su koristili digitalne blizance lanca opskrbe za ažuriranje i korekciju relevantnih parametara sustava lanca opskrbe u stvarnom vremenu, čime su omogućili robusnu optimizaciju u stvarnom vremenu. Na kraju je proveden simulacijski eksperiment prema primjeru. Rezultati eksperimenta pokazuju da ažuriranje relevantnih parametara pomoću digitalnih blizanaca može pomoći menadžerima poduzeća u formuliranju racionalnih planova upravljanja te učinkovito izbjeći problem uskih grla u sustavu lanca opskrbe tortama.

Članak (Raska, P. i ostali, 2021) predstavlja pristupe za smanjenje količine podataka generiranih tijekom simulacijske optimizacije s digitalnim blizancem. Cilj istraživanja bio je testirati i prilagoditi različite metode optimizacije i njihove postavke koje su pogodne za diskretnu simulacijsku optimizaciju u skladu s konceptom Industrije 4.0. Metodika je potvrđena korištenjem aplikacije razvijene za upravljanje izvođenjem paralelnih simulacijskih eksperimenata (s arhitekturom klijent-poslužitelj) s digitalnim blizancem. Predložena metodika smanjuje količinu podataka generiranih tijekom simulacijske optimizacije kombiniranjem različitih pristupa globalne optimizacije, programiranja i tehnika obrade velikih podataka.

Pregled postojećih istraživanja pokazuje da su simulacijski eksperimenti široko prepoznati kao učinkovit pristup za razvoj digitalnih blizanaca. U kontekstu ovog istraživanja, simulacijski eksperiment ima višestruku funkciju, od kojih je primarna verifikacija predložene metodike za razvoj digitalnih blizanaca proizvodnih poslovnih procesa. Simulacijski eksperiment služi kao instrument kojim se procjenjuje koliko digitalni blizanac, razvijen prema predloženoj metodici, uspješno replicira ponašanje stvarnog proizvodnog poslovnog procesa. Međutim, njegova uloga u istraživanju

nadilazi isključivo funkciju validacije na način da je eksperiment ujedno i istraživački alat koji omogućuje komparaciju dvaju pristupa: tradicionalnog modeliranja poslovnih procesa putem simulacije i suvremenog pristupa kroz razvoj digitalnog blizanca.

Eksperiment omogućuje empirijsku usporedbu simuliranih modela i razvijenih digitalnih blizanaca, analizirajući njihove performanse, fleksibilnost i sposobnost reagiranja u stvarnom vremenu. Tako se ne samo potvrđuje ispravnost metodike, već se i demonstrira njezina dodana vrijednost u odnosu na konvencionalne pristupe modeliranju. Na taj se način simulacijski eksperiment koristi kao istraživačka platforma za dokazivanje hipoteze da digitalni blizanac, razvijen prema predloženoj metodici, može vjerodostojno emulirati stvarni proces i podržati prediktivno odlučivanje.

Osim toga, eksperiment doprinosi evaluaciji primjenjivosti metodike na različite proizvodne scenarije (proizvodnja dronova i mobilnih uređaja), čime se potvrđuje njezina fleksibilnost i mogućnost šire primjene u industrijskoj praksi. U tom smislu, simulacijski eksperiment ima i demonstracijsku funkciju jer služi kao pokazni primjer za implementaciju digitalnih blizanaca u stvarnim okruženjima.

Na kraju možemo zaključiti kako simulacijski eksperiment u ovom istraživanju nije korišten samo radi formalne validacije, već je ključni element istraživačkog dizajna kojim se testira, dokazuje i ilustrira primjenjivost razvijene metodike, čime je osigurana veza između teorijskog doprinosa i praktične implementacije.

7 Općenito o digitalnim blizancima

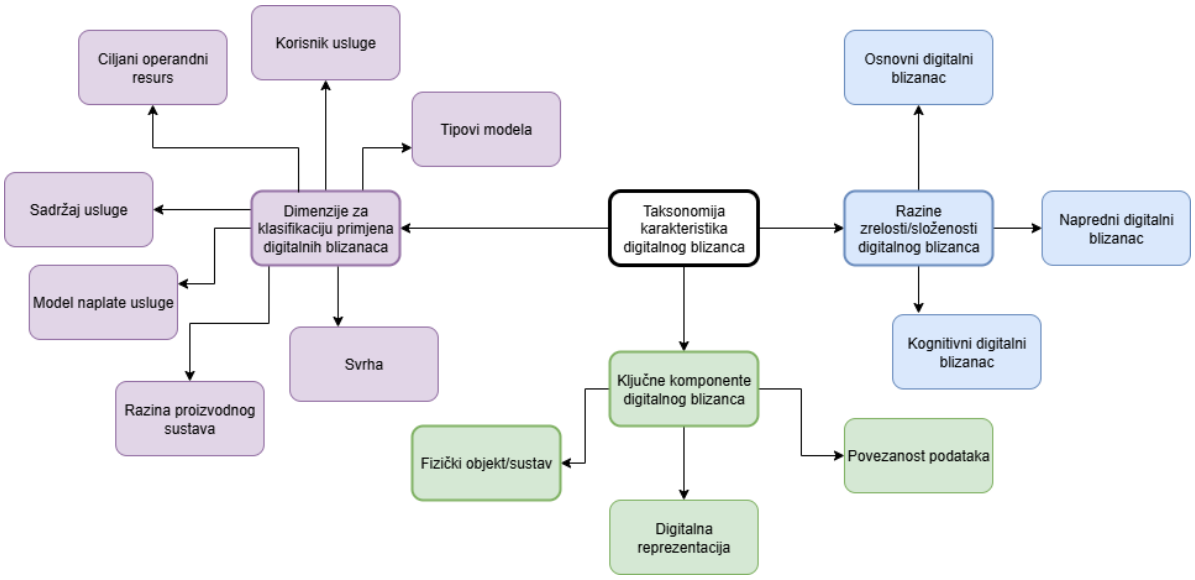
Dumas (2021) kaže kako je digitalni blizanac model objekta ili sustava koji, zajedno s nizom podataka o događajima vezanima uz objekt ili sustav, može točno predvidjeti performanse fizičkog objekta ili sustava tijekom vremena. Digitalni blizanac poslovnog procesa je model poslovnog procesa koji u realnom vremenu integrira aktivnosti od različitih sudionika u sustavu te prevodi sirove podatke u formu koja se može smatrati informacijom za sudionike i podržati donošenje odluka (West, S. i ostali, 2021), (Mallek-Daclin, S. i ostali, 2022), (Meierhofer, J. i ostali, 2020). Digitalni blizanac prikazuje stalnu integraciju fizičkog i virtualnog prostora prije i tijekom rada sustava s trajnom dvosmjernom razmjenom podataka (der Landwehr, M.A. i ostali, 2021) kako bi bio kontinuirano koherentan i usklađen sa svojim stvarnim dvojnikom (Bocciarelli, P. i ostali, 2023). Marmolejo-Saucedo (2020) navodi kako je razvoj digitalnih blizanaca započeo od strane NASA-e, koja je pratila izvedbu satelita i predviđala promjene pomoću simulatora. Kako bi se izgradio digitalni blizanac poslovnog procesa, potrebno je posjedovati domensko znanje o poslovnom procesu kako bi se modelirale aktivnosti i uloge koje su uključene u poslovni proces (Palchunov, D. & Vaganova, A., 2021), (Romero, D. i ostali, 2021), te je potrebno uspostaviti poveznice sa stvarnim poslovnim procesom kojima će se razmjenjivati podaci u realnom vremenu.

Digitalni blizanci nude mogućnost promjene poslovne strategije kako se postojeći proizvodi ili usluge razvijaju, prodaju ili kako se njima upravlja (Nast, B. i ostali, 2022) te spadaju među najvažnije sastavnice najnovije revolucije Industrije 4.0 (Lugaresi, G. & Matta, A., 2021). Digitalni blizanac poslovnog procesa se sastoji od komponenti: modela stvarnog poslovnog procesa, modela blizanca i podataka o poslovnom procesu (Marmolejo-Saucedo, J.A., 2020), (TechTarget, 2023).

Razvojne aktivnosti digitalnog blizanca su obično motivirane poslovnim potrebama koje proizlaze iz, npr., digitalne transformacije, upravljanja poslovnim procesima ili inovacije proizvoda, odnosno, promjenama na tržištu. Motivacija za izradom digitalnog blizanca se može promijeniti tijekom vremena, što može imati utjecaj na dizajn digitalnog blizanca. Osim toga, opažanja tijekom operacija koje izvodi digitalni blizanac mogu implicirati promjene u dizajnu digitalnog blizanca ili ukazivati na nove poslovne prilike koje mogu utjecati na poslovnu motivaciju (Nast, B. i ostali, 2022).

Postoji mnogo prednosti digitalnog blizanca poslovnog procesa, budući da on bilježi sve događaje stvarnog poslovnog sustava te operativne i financijske podatke o sustavu kao i konfiguraciju uređaja (Marmolejo-Saucedo, J.A., 2020).

Taksonomija digitalnog blizanca pomaže kategorizirati i razumjeti različite vrste i primjene digitalnih blizanaca. Postoji nekoliko vrsta (Saif, W. i ostali, 2024), (Van der Valk, H. i ostali, 2020), (Weingram, A. i ostali, 2025), (Galera-Zarco, C. & Papadonikolaki, E., 2023), (Brucherseifer, E. i ostali, 2024), (Aurich, J.C. i ostali, 2025) taksonomija digitalnih blizanaca, ovisno o autorima. Te taksonomije često uzimaju u obzir fizički objekt ili sustav koji se predstavlja, razinu vjernosti i složenosti virtualnog modela te vrste podataka i interakcija koje su uključene, te su stoga uzete u obzir kako bi se napravio pregled taksonomije digitalnih blizanaca (slika 4 u nastavku).



Slika 4 Taksonomija karakteristika digitalnog blizanca

Razine zrelosti/složenosti digitalnog blizanca prema (Uhlenkamp, J.-F. i ostali, 2022)**,** (Weingram, A. i ostali, 2025):

1. Osnovni digitalni blizanac (eng., *Basic Digital Twin*): Jednostavna virtualna reprezentacija fizičkog objekta ili sustava, često korištena za vizualizaciju i osnovno praćenje.
2. Napredni digitalni blizanac (eng., *Advanced Digital Twin*): Uključuje složenije modele, potencijalno obuhvaćajući simulacije temeljene na fizikalnim zakonima, strojno učenje i integraciju podataka u stvarnom vremenu.
3. Kognitivni digitalni blizanac (eng., *Cognitive Digital Twin*): Koristi umjetnu inteligenciju i strojno učenje za analizu podataka, predviđanje budućih stanja, pa čak i automatizaciju donošenja odluka.

Dimenzije za klasifikaciju primjena digitalnih blizanaca prema (Galera-Zarco, C. & Papadonikolaki, E., 2023), (Brucherseifer, E. i ostali, 2024), (Aurich, J.C. i ostali, 2025):

1. Korisnik usluge (eng., *Service Recipient*): Tko koristi digitalnog blizanca (npr. proizvođač, krajnji korisnik…)?
2. Ciljani operandni resurs (eng., *Target Operand Resource*): Koji fizički resurs ili sustav je predstavljen (npr. stroj, proizvodna linija, cijela tvornica)?
3. Sadržaj usluge (eng., *Service Content*): Koje konkretne funkcionalnosti ili uvide pruža digitalni blizanac (npr. praćenje performansi, prediktivno održavanje, optimizacija procesa)?
4. Model naplate usluge (eng., *Service Pricing Model*): Na koji način se usluga digitalnog blizanca nudi i naplaćuje (npr. pretplata, plaćanje po korištenju)?
5. Razina proizvodnog sustava (eng., *Manufacturing System Level*): Za proizvodne primjene, ovo može biti na razini tvornice, stroja ili procesa.
6. Svrha (eng., *Purpose*): Može uključivati analizu, praćenje i upravljanje ili predviđanje.
7. Tipovi modela (eng., *Model Types*): Odnosi se na tip modela korištenog u digitalnom blizancu (npr. „crna kutija“ (eng., *black-box*), „siva kutija“ (eng., *grey-box*), „bijela kutija“ (eng., *white-box*)).

Ključne komponente digitalnog blizanca prema (Botin-Sanabria, D.M. i ostali, 2022):

1. Fizički objekt/sustav (eng., *Physical Object/System*): Stvarni entitet iz stvarnog svijeta koji se predstavlja.
2. Digitalna reprezentacija (eng., *Digital Representation*): Virtualni model fizičkog objekta ili sustava.
3. Povezanost podataka (eng., *Data Connection*): Protok podataka između fizičkog i digitalnog svijeta.

Priložena taksonomija karakteristika digitalnog blizanca iz slike 4 izrađena je s ciljem sustavnog prikaza ključnih dimenzija koje omogućuju klasifikaciju, razumijevanje i usporedbu različitih pristupa u razvoju i primjeni digitalnih blizanaca. Vizualni prikaz integrira tri osnovna koncepta: dimenzije primjene digitalnih blizanaca, razine zrelosti i složenosti te njihove ključne komponente. Ovakva taksonomija doprinosi jasnoći pri određivanju svrhe i opsega pojedinog digitalnog blizanca, omogućujući bolje razumijevanje odnosa između korisničkih potreba, tehničkih značajki i razine sofisticiranosti modela. Korištenjem ovog strukturiranog prikaza moguće je preciznije definirati tip digitalnog blizanca, njegovu funkcionalnost i potencijalnu upotrebu u kontekstu poslovnih procesa.

Postoji nekoliko različitih vrsta digitalnih blizanaca prema (McKinsey&Company, 2024):

1. Proizvodni blizanac, odnosno, digitalni blizanac proizvoda. Ovaj digitalni blizanac može uključivati proizvode u različitim fazama životnog ciklusa, od inicijalnog dizajna koncepta do njegove pune funkcionalnosti. To znači da se podaci o proizvodu dobivaju u stvarnom vremenu kao kada se proizvod servisira.

2. Podatkovni digitalni blizanac. Jedan od ponajboljih primjera podatkovnog digitalnog blizanca su Google Karte koje su digitalni blizanac Zemljine površine. One povezuju podatke o prometu u stvarnom vremenu kako bi se optimizirao, npr., put na posao.

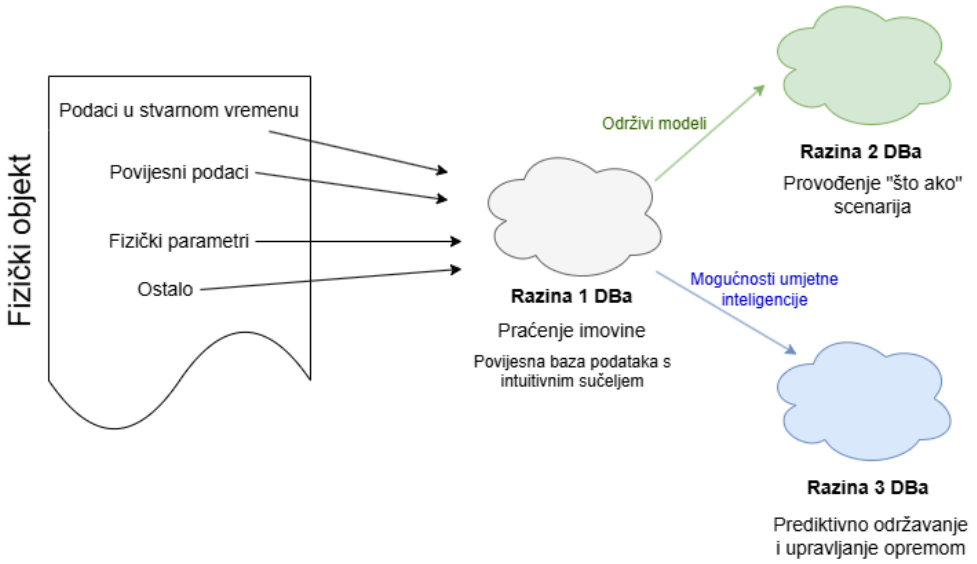
3. Digitalni blizanac organizacijskog sustava, koji prikazuje interakciju između fizičkih i digitalnih procesa, uključujući proizvodne procese, upravljanje lancem opskrbe i poslovanje trgovine.

4. Infrastrukturni digitalni blizanac koji predstavlja fizičku infrastrukturu kao što je autocesta, zgrada ili, npr., stadion.

Digitalni blizanac organizacijskog sustava uključuje digitalnog blizanaca proizvodnog poslovnog procesa. Razlike između digitalnih blizanaca proizvodnih poslovnih procesa i ostalih digitalnih blizanaca jesu u tome što se digitalni blizanci proizvodnih poslovnih procesa usredotočuju na prikaz, nadzor i optimizaciju sekvencijalnih aktivnosti unutar proizvodnog tijeka s ciljem postizanja veće učinkovitosti, pouzdanosti i fleksibilnosti proizvodnje. Njih karakterizira dinamičko povezivanje s, npr., IoT uređajima i proizvodnim strojevima s ciljem simulacije proizvodnog poslovnog procesa i prediktivne analize za donošenje odluka u stvarnom vremenu, a koje su vezane uz proizvodni poslovni proces. Za razliku od drugih vrsta digitalnih blizanaca koji se mogu fokusirati isključivo na proizvod ili prostor (npr. infrastrukturu), digitalni blizanci proizvodnih poslovnih procesa naglasak

stavljaju na poslovnu logiku, slijed aktivnosti i upravljanje resursima u svrhu optimizacije cijele proizvodnje.

Na slici 5 u nastavku se nalaze tri razine digitalnih blizanaca. Na prvoj razini, sustav omogućava praćenje imovine putem povijesne baze podataka, koristeći intuitivno korisničko sučelje. Primjenom održivih modela prelazi se na drugu razinu, koja omogućava izvođenje „što ako“ scenarija. Daljnjim integriranjem mogućnosti umjetne inteligencije, digitalni blizanac doseže treću razinu, čime se omogućava prediktivno održavanje i upravljanje opremom. Ova razina ima ključnu ulogu u industrijskom kontekstu jer ističe prednosti digitalnih blizanaca u unaprjeđenju učinkovitosti i pouzdanosti proizvodnih sustava. Pošto je treća razina najviša razina, tada je ona u fokusu ovog istraživanja.



Slika 5 Razine digitalnih blizanaca

8 Proširenje standardnog pristupa modeliranja poslovnih procesa za digitalne blizance

U ovom poglavlju će se opisati ideja za provedbom doktorskog istraživanja koja se svodi na proširenje postojećeg standardnog pristupa modeliranja poslovnih procesa za digitalne blizance.

Upravljanje poslovnim procesima (BPM) predstavlja učinkovitu metodiku za upravljanje izvedbom procesa koja se koristi kod upravljanja procesno orijentiranim organizacijama (Ubaid, Alaa M & Dweiri, Fikri T., 2020). Autori (Szelągowski, M. & Berniak-Woźny, J., 2024) proveli su sustavni pregled literature s ciljem identificiranja glavnih izazova i ograničenja trenutnih smjerova razvoja BPM-a te definiranja ključnih područja promjena potrebnih za budući BPM koji bi mogao odgovoriti na izazove Industrije 4.0 i nadolazeće Industrije 5.0. Ako se uspješno implementira, BPM može dovesti do povoljnih rezultata jer se fokusira na dizajn završnih (eng., *end-to-end*) procesa (Neufeld, N. & Deo, B., 2021). Naglasak je da BPM koristi poslovne procese kao polazišnu točku i ne koncentrira se samo na pojedinačne događaje te bi također trebao optimizirati upravljanje poslovnim procesima kao cilj, a ne se zadovoljiti samo obradom transakcija (Zhai, Z. & Chen, H., 2010).

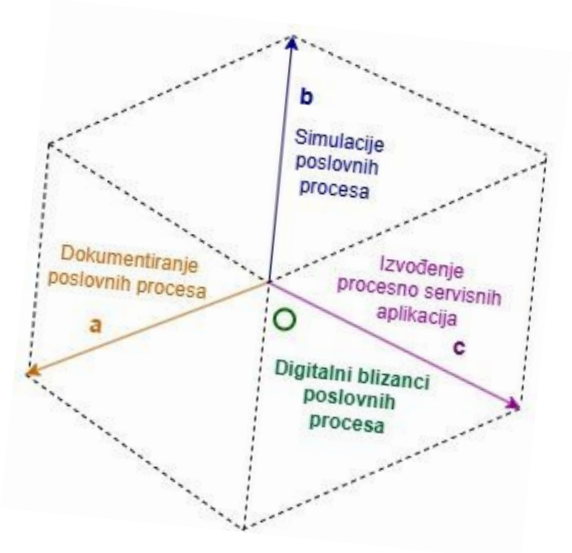
U literaturi su autori (Ubaid, Alaa M & Dweiri, Fikri T., 2020) iskazali da postojeće metodike korištene za upravljanje poslovnim procesima su ili vrlo zastarjele, ne pokrivaju interakciju ljudi s BPM sustavima ili samo djelomično opisuju BPM metodiku. Literatura (Meziani, R. & Saleh, I., 2011) argumentira da postojeće metodike poslovnih procesa ne nude potrebnu fleksibilnost ili agilnost te da su potrebni novi pristupi. Novi pristupi modeliranju poslovnih procesa predloženi su u literaturi (Meziani, R. & Saleh, I., 2011) kako bi se podržao skup povezanih alata koji potiču kolaborativni i inkrementalni dizajn i implementaciju radnih procesa. To se postiže modeliranjem i implementacijom poslovnih procesa u kontinuiranom ciklusu koji prima povratne informacije iz stvarne primjene zadnjih implementiranih procesa.

Neufeld i Deo (Neufeld, N. & Deo, B., 2021) predlažu da bi postojeću metodiku od Dumasa (Dumas, M. i ostali, 2013) trebalo unaprijediti uključivanjem procjene prilagođenosti organizaciji kao prvog koraka prije početka primjene metodike za poboljšanje procesa. Slijedeći primjere prethodnih pokušaja, postojeće metodike upravljanja poslovnim procesima trebale bi se proširiti kako bi odgovarale koracima za razvoj digitalnih blizanaca proizvodnih poslovnih procesa.

Modeliranje poslovnih procesa predstavlja temeljni korak u razumijevanju, optimizaciji i digitalizaciji poslovnih aktivnosti, pri čemu se poslovni procesi dokumentiraju, simuliraju i izvršavaju unutar integriranog okvira. Ova ideja se može vizualizirati kroz simbolički trodimenzionalni prikaz, odnosno prostor s tri međusobno nepovezane osi. Prva os, os *a*, odnosi se na dokumentiranje poslovnih procesa, čime se stvara statički prikaz procesa uz pomoć alata poput Microsoft Visio ili sličnih rješenja za modeliranje. Druga os, os *b*, obuhvaća simulaciju poslovnih procesa, pri čemu se definiraju ključni parametri poput trajanja aktivnosti i alokacije resursa, omogućujući analizu dinamike procesa i optimizaciju izvedbe. Treća os, os *c*, predstavlja izvršenje

poslovnih procesa, koje uključuje razvoj procesno servisnih, odnosno, orijentiranih aplikacija. U ovoj fazi dodaju se složeniji elementi poput podatkovnog modela, izrade formi, definiranja poslovnih pravila te drugih relevantnih parametara koji omogućuju operativnu provedbu procesa kroz aplikacijska rješenja.

Na slici 6 u nastavku je simbolički, trodimenzionalni prikaz položaja digitalnih blizanaca poslovnih procesa u odnosu na osi koje predstavljaju dokumentiranje poslovnih procesa, simulacije poslovnih procesa i izvođenje procesno servisnih aplikacija.



Slika 6 Proširenje postojećeg standarada modeliranja poslovnih procesa

Prema prethodnom prikazu, unutar simboličkog, trodimenzionalnog prostora nalazi se digitalni blizanac poslovnog procesa, koji integrira sve tri razine: dokumentiranje poslovnih procesa, simulaciju poslovnih procesa i izvođenje procesno servisnih aplikacija. Digitalni blizanac omogućava cjelovit prikaz poslovnog procesa u njegovim različitim fazama te je tako sadržan unutar sve tri osi, odnosno, komponente. Doprinos ovog pristupa očituje se u podizanju načina upravljanja poslovnim procesima na novu, složeniju razinu čime se unaprjeđuje sposobnost organizacije da pravovremeno odgovori na suvremene izazove i zahtjeve poslovanja. Sukladno navedenom, doprinos ovog istraživanja je unaprjeđenje metodike modeliranja poslovnih procesa kako bi se razvila metodika za razvoj digitalnih blizanaca poslovnih procesa.

9 Studija slučaja: Inovacijski park Biel/Bienne

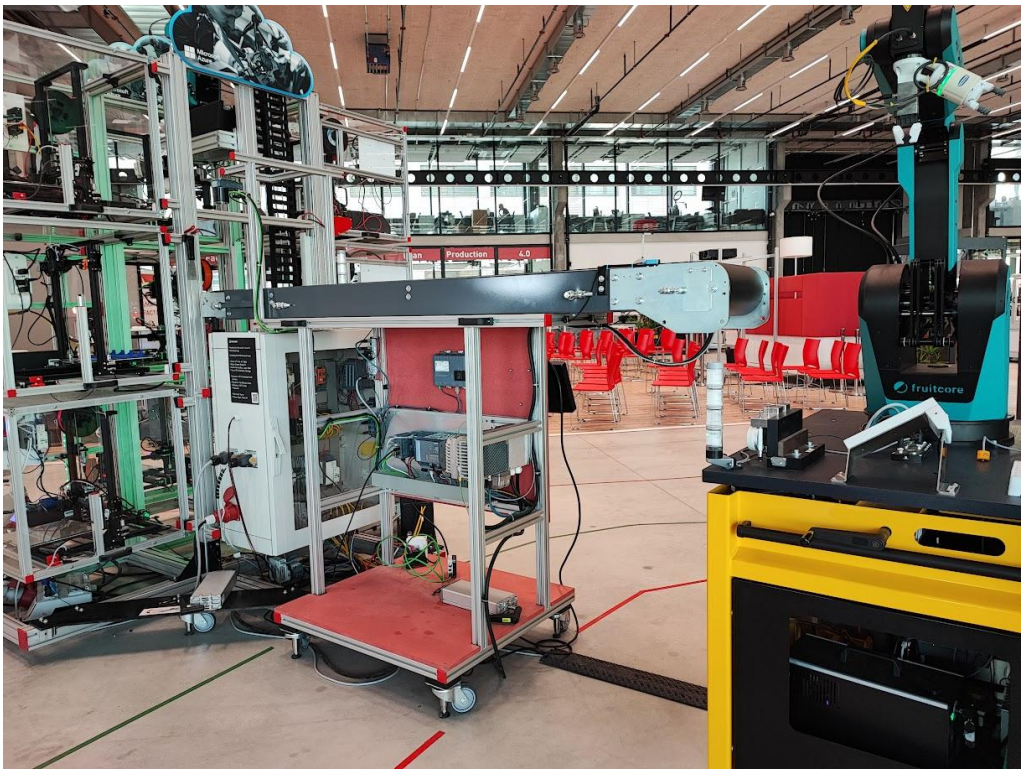
9.1 Opis studije slučaja

Za potrebe izrade doktorskog rada, bilo je potrebno naći poduzeće koje se bavi proizvodnjom, a da je proizvodnja barem u nekom segmentu podržana uređajima Interneta stvari. Uređaji Interneta stvari korišteni u proizvodnji bi bile, primjerice, kamere ili senzori. Za prvi slučaj korištenja je tako uzet Inovacijski park Biel/Bienne u Švicarskoj[9]. Inovacijski park Biel/Bienne privatna je švicarska neprofitna organizacija koja provodi i podupire industrijski povezana i primijenjena istraživanja i razvoj. Kao dio nacionalne i međunarodne mreže Švicarske Zaklade za Inovacije, cilj Inovacijskog parka je generirati ulaganja u istraživanja iz inozemstva, promicati švicarske inovacije i *start-up*-ove te ubrzati prevođenje istraživačkih rezultata u tržišne proizvode (*Switzerland Innovation Park Biel/Bienne*, 2024.).

Unutar Inovacijskog parka posluju centri za istraživanje i razvoj koji pokazuju i demonstriraju kako praktično primijeniti nove tehnologije te na taj način zatvoriti jaz između istraživanja i industrije (*Switzerland Innovation Park Biel/Bienne*, 2024.). Kako bi omogućili primjenu stečenog znanja, u Inovacijskom parku djeluje Švicarska Pametna Tvornica (eng., *Swiss Smart Factory*) koja je međunarodno priznat centar kompetencije u istraživanju orijentiranom na primjenu Industrije 4.0. *Swiss Smart Factory* se etablirala kao prva platforma za testiranje i demonstraciju za Industriju 4.0. u Švicarskoj. S vremenom je postala „supermarket" ideja gdje su tvrtke svih veličina dobrodošle kako bi naučile i pronašle koja tehnologija odgovara njihovim potrebama. Osim toga, kako sve više inovacijskih projekata dolazi na tržište, tako se i infrastruktura za testiranje i demonstraciju kontinuirano ažurira (*Swiss Smart Factory*, 2024). Osim Pametne tvornice, u Inovacijskom parku, u području istraživanja djeluju još Tehnološki centar za zdravlje (eng., *Swiss HealthTech Center*), Centar za unaprijeđenu proizvodnju (eng., *Swiss Advanced Manufacturing center*) i Tehnološki centar za baterije (eng., *Swiss Battery Technology Center*). Tehnološki centar za zdravlje pruža okruženje za vođenje istraživačkih i inovacijskih projekata iz sektora zdravstvenih tehnologija. Centar za unaprijeđenu proizvodnju nudi stručnu podršku za primijenjena istraživanja u naprednim proizvodnim tehnologijama s fokusom na aditivnu proizvodnju baziranoj na metalima koja uključuje lasere. Tehnološki centar za baterije nudi korisnicima *end-to-end* razvoj sustava za pohranu energije. Slike 7 i 8 u nastavku prikazuju postrojenje u Pametnoj tvornici u Švicarskoj.

[9] https://www.sipbb.ch/en/, poveznica na službenu stranicu Inovacijskog parka Biel, pristupano 17.07.2024.

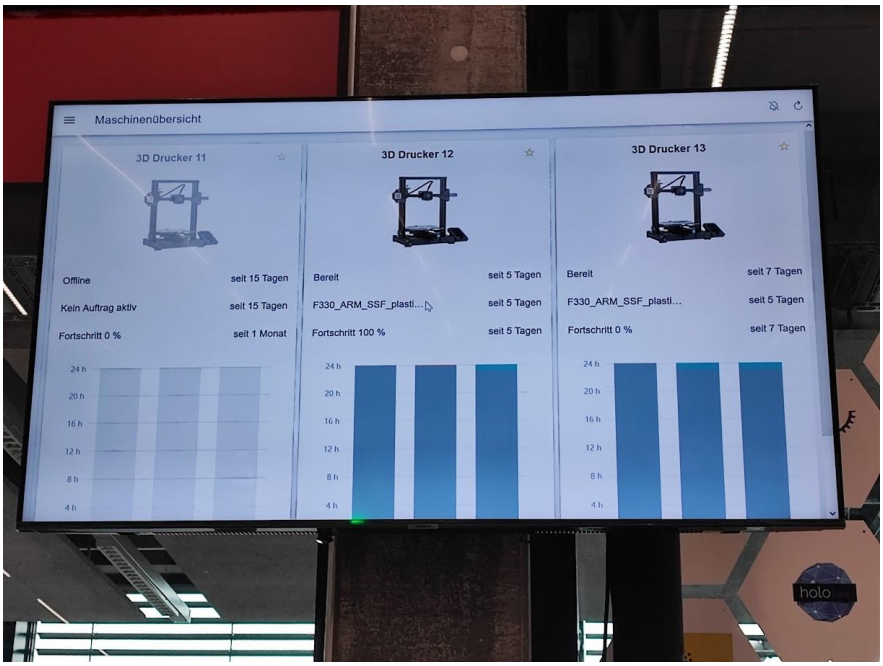
Slika 7 Postrojenje u tvornici



Slika 8 Pokretna traka u tvornici

9.2 Opis poslovnog procesa proizvodnje dronova

Poslovni proces prvog proizvodnog poduzeća, Švicarskog Inovacijskog parka Biel/Bienne će biti opisan u ovom poglavlju i to je korak kada se radi analiza proizvodnog poslovnog procesa. Za potrebe izrade digitalnog blizanca proizvodnog poslovnog procesa, uzet je proces proizvodnje dronova. Proces se izvodi na nekoliko radnih postaja unutar Inovacijskog parka, a započinje zaprimanjem poruke o narudžbi iz ERP sustava. Proces narudžbi prati se na ekranu koji je prikazan na slici 9 ispod.



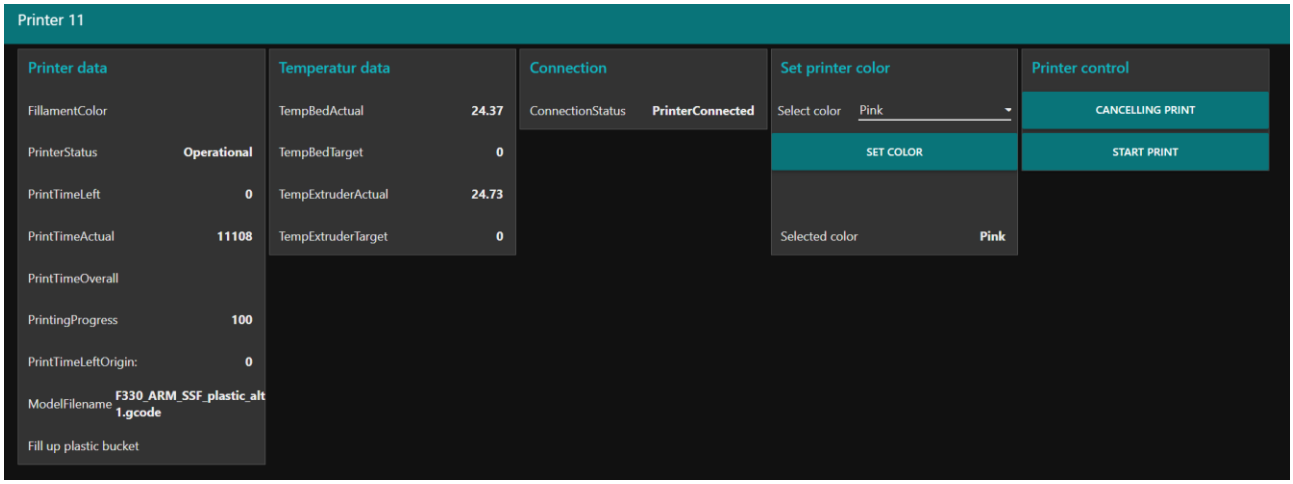
Slika 9 Praćenje procesa narudžbi

U zaprimljenoj poruci pišu informacije o dijelu proizvoda kojeg treba proizvesti. Primjer zaprimljene narudžbe izgleda ovako:

"adress":{"userName":"Strom Max e.G.","userEmail":"info@strom-max.de","userStreet":"Raiffeisenstraße 5","userKanton":"","userZip":"4054","userCity":"Basel"}, "product":[{"OrderID":20405,"product":"0101","edition":"Entry","priceTotal":159,"co2Total":1090.5,"userDelivery":"ECO","productionLocation":"CH","colorBase":"black", "colorBranch":["pink","red","green","yellow"],"logo":"00","engravingText":"","amount":1}]}

Nakon zaprimljene informacije o dijelu proizvoda kojeg treba proizvesti, radnik prema vlastitom iskustvu i nahođenju odlučuje na kojem printeru će pustiti proizvodnju. Ako na odabranom printeru nema dovoljno materijala za printanje dijela, svjetlo na vratima printera će svijetliti crveno što je znak radniku da treba zamijeniti spremnik s materijalom za printanje. Svjetlo je spojeno sa senzorom na printeru. Senzor je okrenut vertikalno prema roli spremnika (*filamenta*) i mjeri da li je udaljenost mala (nit prolazi tamo kamo je usmjeren laser) ili je veća (odbija se od zida printerovog kućišta). Senzor usmjeri laser prema objektu, taj se val svjetla reflektira na drugu leću unutar senzora. Pri tome se onda ovisno o udaljenosti mijenja kut, i na kraju prema tom kutu senzor triangulira stvarnu udaljenost. Ako nema dovoljno materijala, radnik zamijeni spremnik s materijalom i kreće printanje, u suprotnom printanje kreće odmah. Radnik preko upravljačke ploče Node-RED-a bira opciju

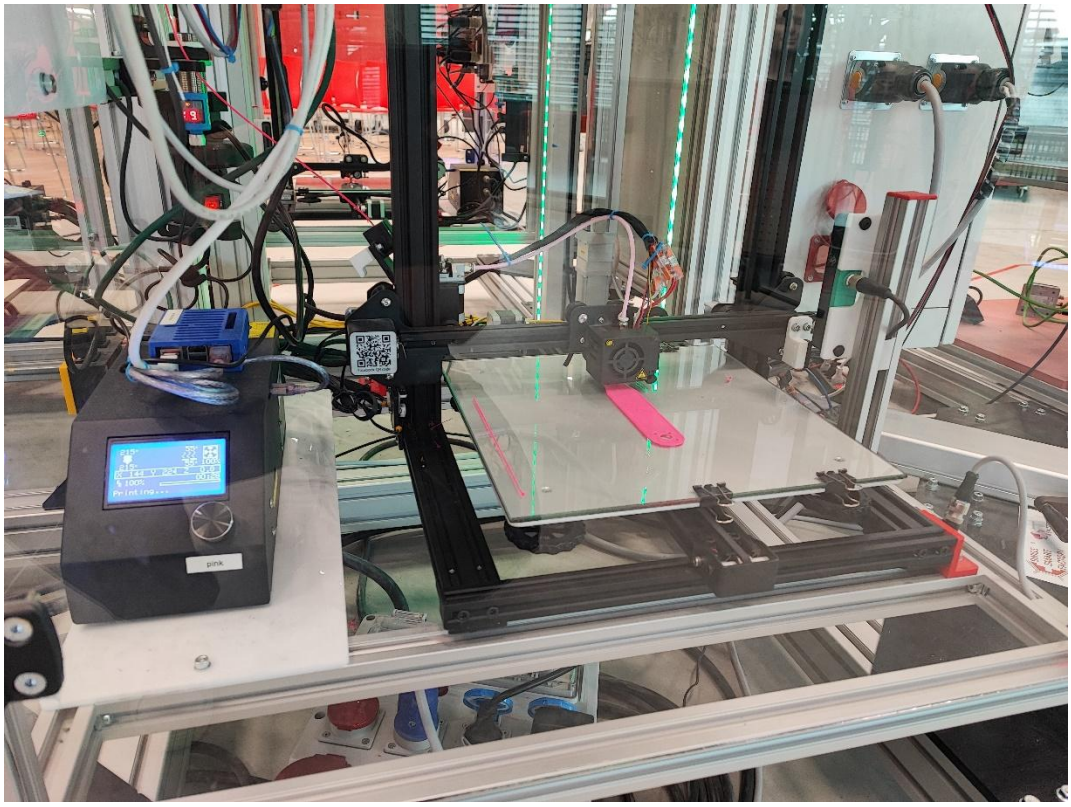
printanja i printanje započinje. Node-RED je vizualni alat za razvoj aplikacija, dizajniran za spajanje hardverskih uređaja, API-ja i mrežnih usluga na jednostavan način. Baziran je na JavaScript-u i razvijen je na platformi Node.js. Node-RED omogućava korisnicima da kreiraju tijekove podataka (eng., *flows*) povlačenjem i povezivanjem čvorova (eng., *nodes*) u grafičkom sučelju (Node-RED, 2024). Glavna primjena Node-RED-a je integracija i automatizacija IoT uređaja, obrada podataka u realnom vremenu te orkestracija različitih servisa u složenim sustavima. Slika upravljačke ploče odakle kreće printanje se nalazi u nastavku.



Slika 10 Prikaz upravljačke ploče iz NodeRED-a

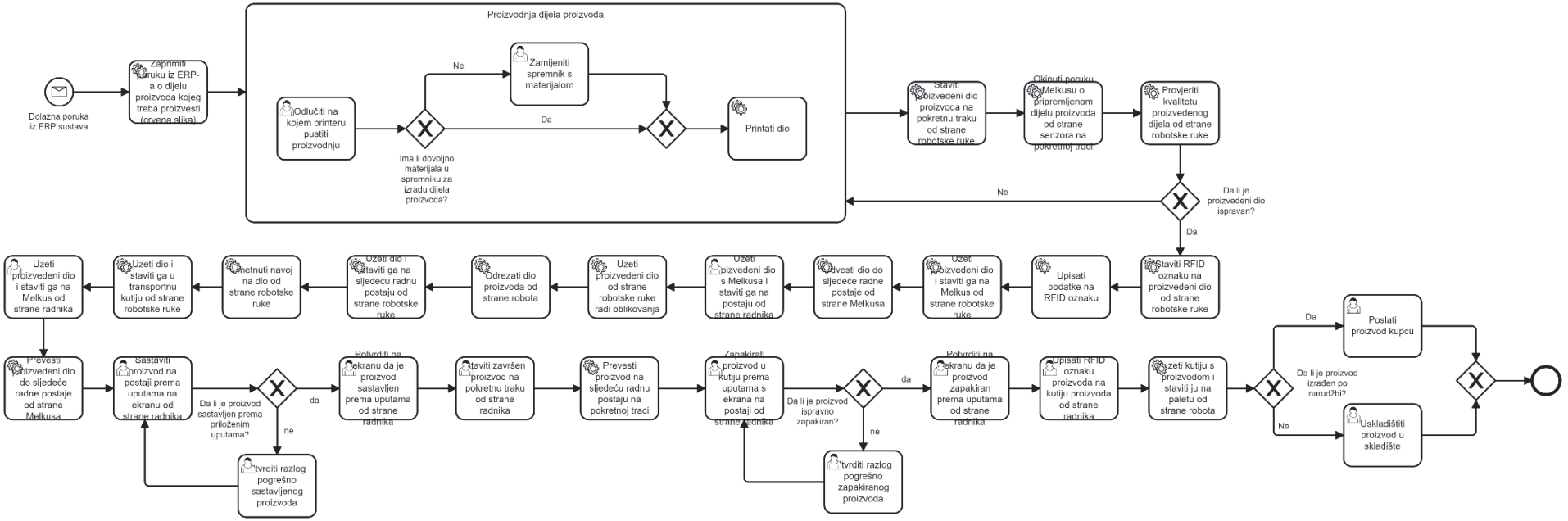
Nakon što radnik inicira printanje, printer se kreće zagrijavati i počinje printati dio, a dio je krilo drona. Zagrijavanje printera traje otprilike 20 min, a sam proces printanja oko 1,5 sat. Primjer kako izgleda printanje krila drona nalazi se na slici 11 u nastavku. Dok printer završi printanje dijela krila drona, senzor na vratima printera svijetli zeleno što je znak da je printer spreman za printanje novog dijela. Na radnoj stanici robotska ruka uzima isprintani dio proizvoda (drona) i stavlja ga na pokretnu traku. Dok je dio na pokretnoj traci, okida se okidač preko senzora te se šalje poruka pokretnom robotu (nadalje Melkusu) o pripremljenom dijelu proizvoda. Robotska ruka provjerava kvalitetu proizvedenog dijela skeniranjem dijela. Ako je proizvedeni dio u redu, proizvodnja se nastavlja. Ako proizvedeni dio nije kvalitetom u redu, novi dio se kreće proizvoditi. Na ispravan proizvedeni dio robotska ruka stavlja RFID oznaku te se upisuju podaci na RFID oznaku. Robotska ruka proizvedeni dio stavlja na Melkus. Melkus prevozi dio do sljedeće radne postaje. Radnik uzima proizvedeni dio s Melkusa i stavlja ga na postaju. Robotska ruka uzima dio i oblikuje ga. Oblikovanje dijela podrazumijeva odrezivanje dijela proizvoda. Robotska ruka uzima dio i stavlja ga na sljedeću radnu postaju. Robotska ruka umeće navoj na dio. Nadalje, robotska ruka uzima dio i stavlja ga u transportnu kutiju. Radnik uzima dio i stavlja ga na Melkusa radi transporta. Melkus prevozi proizvedeni dio do sljedeće radne postaje. Radnik sastavlja proizvod od dijelova na radnoj postaji prema uputama koje ga navode putem ekrana. Tijekom sastavljanja proizvoda, radnik mora na ekranu potvrđivati da je napravio svaki predloženi korak te, u konačnici, potvrđuje da je proizvod sastavljen prema uputama. Radnik stavlja završen proizvod na pokretnu traku te se proizvod prevozi na pokretnoj traci do sljedeće radne postaje. Radnik pakira proizvod u kutiju prema uputama na ekranu na radnoj postaji. Nakon što je proizvod zapakiran, radnik mora potvrditi na ekranu da je proizvod ispravno zapakiran prema uputama. Radnik upisuje RFID oznaku proizvoda na kutiju proizvoda. Robot, odnosno, robotska ruka uzima kutiju s proizvodom i stavlja ju na paletu s drugim kutijama proizvoda. Radnik provjerava da li je proizvod izrađen po narudžbi. Ako je, proizvod se šalje direktno

kupcu. Ako proizvod nije izrađen po narudžbi, skladišti se do prodaje. Tu se proizvodni proces, koji uključuje sastavljanje proizvoda i njegovo pakiranje, završava.



Slika 11 Printanje krila drona

Na slici 12 u nastavku nalazi se poslovni proces proizvodnje drona iz Inovacijskog parka Biel napravljen u alatu Camunda Modeler i to je korak u kojem se mapiraju aktivnosti proizvodnog poslovnog procesa.

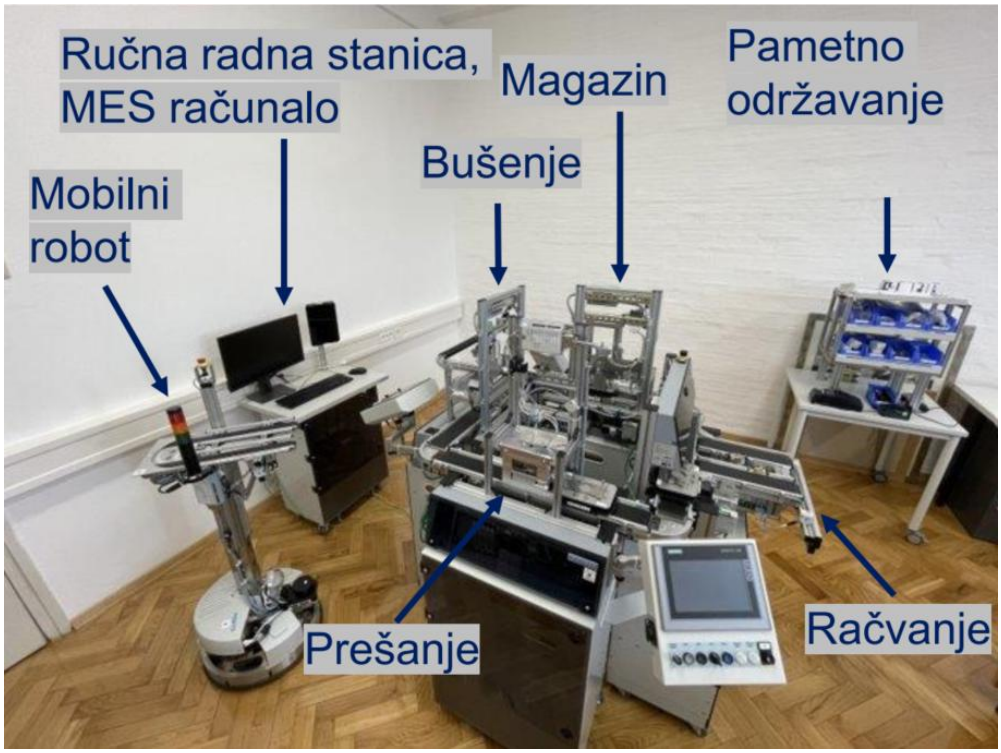


Slika 12 Dijagram procesa AS IS Inovacijskog parka Biel

10 Studija slučaja: Fakultet strojarstva i brodogradnje (FSB) Sveučilišta u Zagrebu

10.1 Opis studije slučaja

Fakultet strojarstva i brodogradnje (FSB) Sveučilišta u Zagrebu „vodeća je znanstveno-istraživačka, obrazovna i stručna institucija u Republici Hrvatskoj u području strojarstva, brodogradnje, zrakoplovnog inženjerstva te, u novije vrijeme, jedna od vodećih u mehatronici i robotici. FSB pruža izvrsnost u inženjerskim znanostima obrazujući buduće inženjere u području visokotehnoloških industrija koje zahtijevaju široko znanje.“ (Fakultet strojarstva i brodogradnje, 2024)
U sklopu FSB-a radi 54 laboratorija te 42 katedre i 14 zavoda, među kojima se nalazi i Zavod za industrijsko inženjerstvo u kojem se nalazi „Pametna tvornica“ s malom proizvodnom linijom koja se koristi za potrebe učenja studenata. Ova proizvodna linija simulira proizvodnju mobilnih uređaja. Sama proizvodna linija je nabavljena u sklopu projekta Zavoda za industrijsko inženjerstvo i Katedre za upravljanje proizvodnjom pod vodstvom dr.sc. Mire Hegedića, mag.ing. Pametna tvornica za učenje je modularna tvornica za učenje, spremna za Industriju 4.0 i usmjerena je prema potrebama tržišta. Nadalje, financirana je sredstvima Europske unije, a u rad je puštena u veljači 2022. Trenutačno se koristi za obrazovanje zaposlenika te studenata preddiplomskog, diplomskog i doktorskog studija.
Za potrebe izrade ovog doktorskog rada, Fakultet strojarstva i brodogradnje je dopustio doktorandici prikupljanje podataka o proizvodnji. Sustav rada unutar Pametne tvornice je prikazan na slici 13 u nastavku.

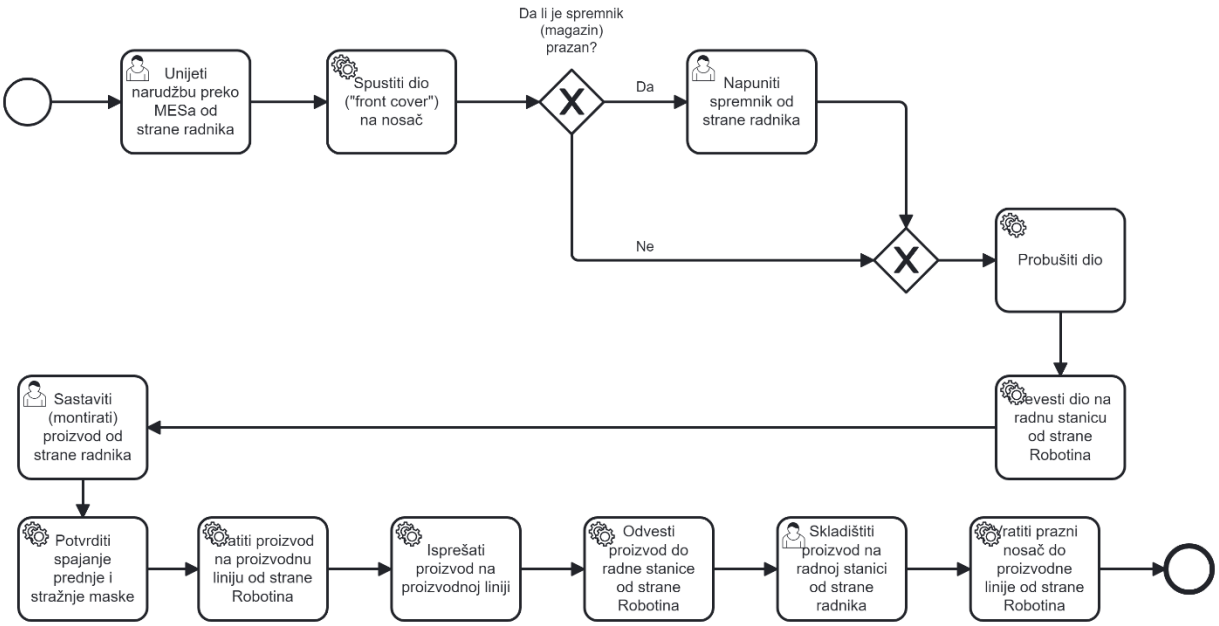


Slika 13 Sustav rada unutar Pametne tvornice (Fakultet strojarstva i brodogradnje, 2022)

Na prethodnoj slici možemo vidjeti kako se proizvodna linija sastoji od sljedećih aktivnosti: magazin, bušenje, prešanje, račvanje i mobilnog robota (Robotino). Ručnu radnu stanicu, na kojoj se nalazi računalo s instaliranom MES aplikacijom koristi radnik, isto kao što se radnik nalazi i za stanicom za pametno održavanje gdje se nalaze dijelovi za proizvodnju.

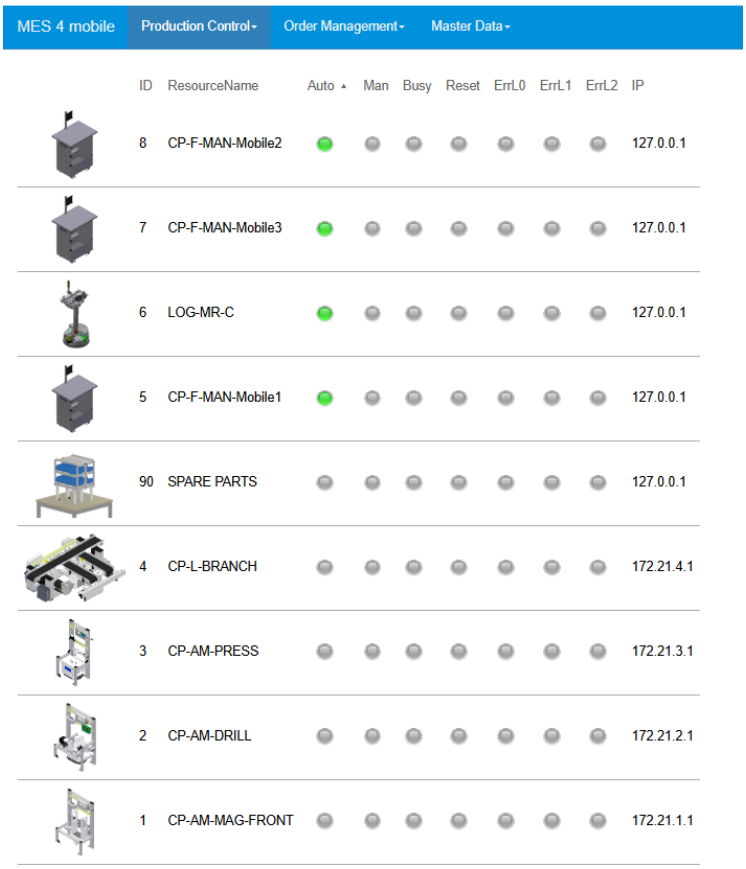
10.2 Opis poslovnog procesa proizvodnje mobilnih uređaja

U ovom poglavlju će biti opisan poslovni proces Pametne tvornice na Fakultetu strojarstva i brodogradnje Sveučilišta u Zagrebu i to je korak kada se radi analiza proizvodnog poslovnog procesa. U Pametnoj tvornici se simulira proizvodnja mobilnih uređaja. Proizvodnja se sastoji od postavljanja prednjeg dijela maske (eng., *front cover*) mobilnog uređaja, bušenja komponente, postavljanja stražnjeg dijela maske (eng., *back cover*) na prednji dio i prešanja proizvoda. Bitno je za istaknuti kako se radi o proizvodnoj liniji za učenje, odnosno, školskoj proizvodnoj liniji. Proces započinje unosom narudžbe preko sustava MES od strane radnika, pri čemu radnik može biti student ili djelatnik Fakulteta. MES (eng. *Manufacturing Execution System*) ili sustav izvršne proizvodnje je softverski sustav koji upravlja i nadzire procese proizvodnje, odnosno pretvaranje sirovine u konačne proizvode. MES u stvarnom vremenu prati sve faze proizvodnje, od narudžbe do isporuke. Nakon primitka poruke o proizvodnji: koji dio se proizvodi, koje boje, u kojoj količini, na proizvodnoj liniji se spušta komponenta, prednji dio maske, na nosač linije. Ako u spremniku/magazinu nema dovoljno dijelova za proizvodnju, radnik će primiti obavijest putem HMI zaslona magazina i unutar kartice „Resursi“ u MES sustavu o praznom spremniku kojeg treba napuniti. U tom spremniku se nalaze dijelovi poput prednjeg dijela maski. Ako spremnik/magazin nije prazan, proizvodnja se redovno izvršava. Prazan spremnik radnik puni ako nema dovoljno dijelova za proizvodnju. Na proizvodnoj liniji se buši komponenta, odnosno, dio za proizvod koji se izrađuje. Prijenos palete s proizvodne linije na Robotino odvija se automatski preko modula "grananje/račvanje". Prijenos se odvija pomoću remenja na pokretnim trakama koje se nalaze na modulu "račvanje" i na Robotinu. Robotino, pokretni robot, prevozi dio do sljedeće radne stanice radi sastavljanja. Na sljedećoj radnoj stanici radnik sastavlja, odnosno, montira proizvod. Ručnu radnu stanicu i Robotino povezuje čovjek. Radnik uzima iz kontejnera na radnoj stanici stražnji dio maske i stavlja ga na prednji dio maske. Radnik vraća paletu s proizvodom na Robotina prema danim uputama koje se prikazuju u modulu "*Manual workstation*" MES sustava. Prilikom preuzimanja i vraćanja palete s proizvodom čovjek vrši interakciju s Robotinom i MES sustavom preko sučelja na Robotinu. Robotino odvozi proizvod nazad na proizvodnu liniju. Prijenos s Robotina na proizvodnu liniju vrši se ponovno automatski preko modula "račvanje". Robotino se uz pomoć senzora pozicionira na modul „račvanje“ te se pomoću pokretnih traka prenosi paleta na proizvodnu liniju. Na pametnoj proizvodnoj liniji se proizvod preša. Proizvod s proizvodne linije na Robotina ponovno dolazi automatski pomoću modula „račvanje“. Isprešani proizvod Robotino prevozi do sljedeće radne stanice radi skladištenja. Radnik skladišti proizvod na radnoj stanici. Robotino vraća prazni nosač do proizvodne linije.
Kako vizualno izgleda proces pametne proizvodnje na FSB-u, moguće je vidjeti i na dijagramu slike 14 u nastavku, koji predstavlja korak mapiranja aktivnosti proizvodnog poslovnog procesa.



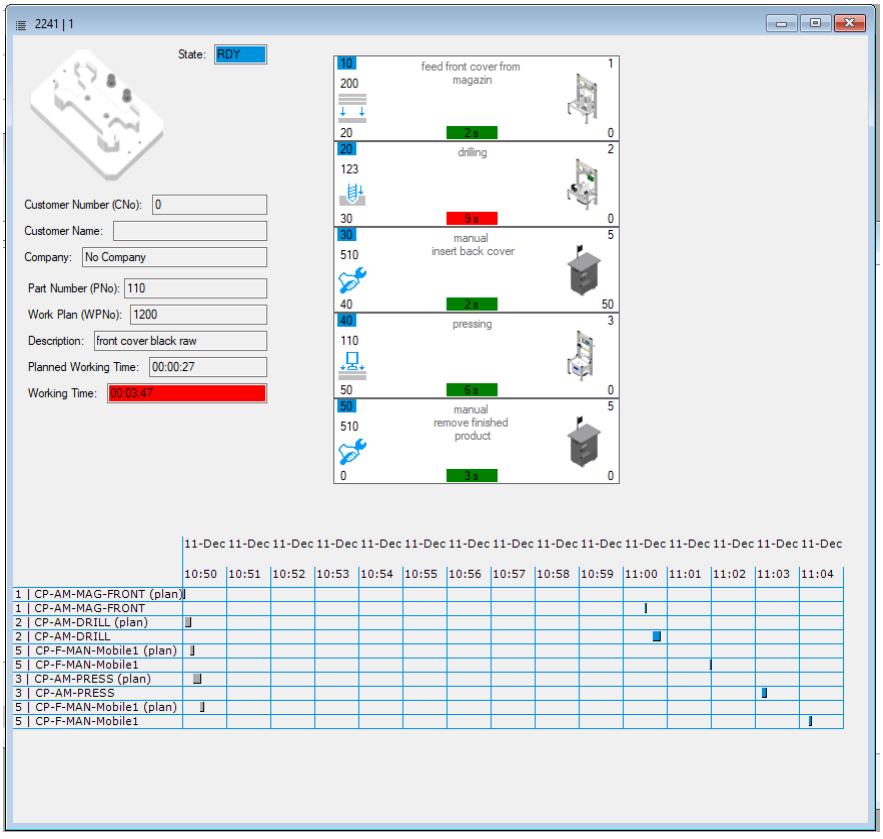
Slika 14 Dijagram procesa AS IS „Pametne tvornice“ na FSB-u

Stanje svakog stroja koji se nalazi u proizvodnoj liniji je moguće pratiti putem MES sustava kao što je vidljivo na slici 15 u nastavku.



Slika 15 Stanje strojeva u proizvodnoj liniji

Na slici 16 u nastavku je vidljivo stvarno trajanje aktivnosti iz procesa koje se prati putem MES sustava. Ovaj podatak o trajanju aktivnosti se koristi kako bi se napravio simulirani model proizvodnog poslovnog procesa kojim se potvrđuje da digitalni blizanac proizvodnog poslovnog procesa odgovara simuliranom modelu, odnosno, da nema razlike u izvođenju proizvodnog digitalnog blizanca u odnosu na simulirani model, osim što se simulirani model temelji na povijesnim podacima, poput ovih, prikupljenih u prošlosti iz stvarnog poslovnog procesa, a digitalni blizanac komunicira u realnom vremenu s poslovnim procesom te je obogaćen stvarnim podacima čime se podupire donošenje operativnih odluka na vrijeme. Ukupno radno vrijeme proizvodnog procesa traje nešto više od 3 minute i, iako je označeno crvenom bojom, koja signalizira kašnjenje, do „kašnjenja“ dolazi zbog vremena koje treba Robotinu da preveze proizvod na nosaču između postaja te vremena koje treba radniku da makne proizvod s nosača i vrati ga na nosač. Drugim riječima, ukupno radno vrijeme traje duže od zbroja trajanja aktivnosti sa slike zato što se u tom ukupnom vremenu nalazi vrijeme kretanja Robotina i ljudski rad, koji se kao takvi ne vide na slici i ne evidentiraju se u sustavu, ali se događaju unutar procesa. Tako spuštanje dijela (eng., *front covera*“) iz magazina na nosač traje 2 sekunde. Bušenje traje 9 sekundi, ručno sastavljanje proizvoda od prednjeg i stražnjeg poklopca traje 2 sekunde, prešanje traje 6 sekundi, uklanjanje završenog proizvoda, odnosno, skladištenje proizvoda na radnoj stanici traje 3 sekunde.



Slika 16 Trajanje proizvodnje po aktivnostima procesa iz MES sustava

11 Koraci razvoja digitalnog blizanca proizvodnog poslovnog procesa

Prema (Dumas, M. i ostali, 2013) životni ciklus modeliranja poslovnih procesa se sastoji od faza:

1. Identifikacija procesa
2. Otkrivanje procesa
3. Analiza procesa
4. Redizajn procesa
5. Implementacija procesa
6. Praćenje procesa.

Identifikacija procesa je faza u kojoj se postavlja poslovni problem. U ovoj fazi se identificiraju, razgraničavaju i međusobno povezuju adresirani procesi koji su relevantni za postavljeni poslovni problem. Ishod identifikacije procesa je nova ili ažurirana arhitektura procesa, koja daje cjelokupnu sliku procesa u organizaciji i njihove veze. Ova arhitektura se zatim koristi za odabir procesa ili skupa procesa za upravljanje kroz preostale faze životnog ciklusa.

Otkrivanje procesa (često se naziva i *modeliranje trenutnog (as-is) procesa*) je faza u kojoj se trenutno stanje svakog od relevantnih procesa obično dokumentira u formi jednog ili nekoliko modela procesa kakvi oni jesu.

Problemi povezani s trenutnim procesom su identificirani, dokumentirani i, gdje god je to moguće, kvantificirani korištenjem mjera učinaka u fazi analize procesa. Ishod faze analize procesa je strukturirana kolekcija pitanja. Ovim problemima se daje prioritet temeljem njihovog potencijalnog utjecaja i procijenjenog napora potrebnog za njihovo rješavanje.

Redizajn procesa (često se zove i *poboljšanje procesa*) ima za cilj identificirati promjene u procesu koje bi pomogle u rješavanju problema identificiranih u prethodnoj fazi i omogućile organizaciji da ispuni svoje ciljeve. U tu se svrhu analizira i uspoređuje više opcija da se vidi koja bi imala više učinaka. Ishod ove faze je obično budući (*to-be*) model procesa.

Implementacija procesa je faza u kojoj se pripremaju i provode promjene potrebne za prelazak s trenutnog (*as-is*) procesa na budući (*to-be*) proces. Implementacija procesa pokriva dva aspekta: upravljanje organizacijskim promjenama i automatizaciju. Upravljanje organizacijskim promjenama odnosi se na skup aktivnosti potrebnih za promjenu načina rada svih sudionika uključenih u proces. Automatizacija procesa odnosi se na razvoj i implementaciju IT sustava koji podržavaju budući (*to-be*) proces.

Nakon što se redizajnirani proces pokrene, prikupljaju se i analiziraju relevantni podaci kako bi se utvrdilo koliko dobro proces radi s obzirom na mjere izvedbe i ciljeve izvedbe. Identificiraju se uska grla, pogreške koje se ponavljaju ili odstupanja u odnosu na planirano ponašanje i poduzimaju se korektivne radnje. Sve ovo se provodi u fazi praćenja procesa (Dumas, M. i ostali, 2013).

Prema (Asana, 2024) metodika za upravljanje poslovnim procesima uključuje sljedeće korake:

1. Analiza: Analiza poslovnih procesa i njihovo mapiranje od početka do kraja izvođenja.
2. Modeliranje: Izrada modela poslovnog procesa u njegovom trenutnom stanju.
3. Implementacija: Puštanje modela poslovnog procesa u pogon.
4. Nadzor: Praćenje uspješnosti izvedbe modela poslovnog procesa kroz definirane metrike.
5. Optimizacija: Optimizacija modela poslovnog procesa uklanjanjem identificiranih neučinkovitosti.

U okviru razmatranja različitih pristupa upravljanju poslovnim procesima, modeli koje predlažu Dumas i ostali te Asana su uzeti u obzir zbog njihove jasne strukture, široke prihvaćenosti u znanstvenoj i stručnoj zajednici te praktične primjenjivosti u modeliranju i optimizaciji procesa. Ovi pristupi obuhvaćaju ključne faze životnog ciklusa poslovnog procesa od identifikacije do optimizacije što omogućuje sustavnu i iterativnu izgradnju digitalnog blizanca proizvodnog poslovnog procesa od inicijalnog modeliranja do konačne evaluacije i optimizacije. Za razliku od brojnih drugih teorijskih pristupa, Dumasov okvir se odlikuje ravnotežom između formalnosti i primjenjivosti, uz naglasak na BPMN standard kao univerzalni jezik za modeliranje, što je posebno važno u kontekstu razvoja digitalnog blizanca proizvodnog poslovnog procesa koji mora biti razumljiv i precizan. Oba pristupa imaju sličnosti, a cilj je isti: identificirane probleme u trenutnom poslovnom procesu otkloniti te optimizirati budući poslovni proces. Metodiku za upravljanje poslovnim procesima treba proširiti na način da zadovoljava korake izrade digitalnog blizanca proizvodnog poslovnog procesa.
Koraci razvoja digitalnog blizanca proizvodnog poslovnog procesa su tako:

1. Analiza: Analizirati proizvodni poslovni proces i evidentirati sve aktivnosti proizvodnog poslovnog procesa, odnosno, mapirati ga od početka do kraja izvođenja.
2. Simulacija: Izraditi procesnu simulaciju kako bi se omogućila usporedba njezinih rezultata s izvođenjem proizvodnog digitalnog blizanca te utvrdila eventualna odstupanja u njegovom radu.
3. Modeliranje: Razviti digitalnog blizanca proizvodnog poslovnog procesa s obzirom na njegovo stanje.
    a. Ugraditi složenije poslovno pravilo, element umjetne inteligencije, u model digitalnog blizanca.
    b. Ugraditi element za komunikaciju digitalnog blizanca proizvodnog poslovnog procesa s vanjskim sustavom.
        i. Izraditi proizvodnog digitalnog blizanca u alatu za izradu procesno servisnih arhitektura.
        ii. Izraditi Python skriptu za komunikaciju vanjskog sustava s modelom digitalnog blizanca.
4. Implementacija: Pustiti digitalnog blizanca proizvodnog poslovnog procesa u pogon.
5. Nadzor: Pratiti izvođenje proizvodnog digitalnog blizanca.
    a. Pratiti uspješnost izvedbe digitalnog blizanca proizvodnog poslovnog procesa temeljem stvarnih podataka prikupljenih iz vanjskog sustava.
    b. Usporediti rezultate simuliranog modela proizvodnog poslovnog procesa s modelom digitalnog blizanca proizvodnog poslovnog procesa.
6. Optimizacija: Optimizirati rad digitalnog blizanca proizvodnog poslovnog procesa uklanjanjem identificiranih neučinkovitosti.

Usporedba tradicionalnog pristupa modeliranja poslovnih procesa i potrebe što bi nova metodika za razvoj proizvodnih digitalnih blizanaca trebala uključivati je prikazana u tablici 2 u nastavku.

<table>
<tr><th>Tradicionalni pristupi modeliranja</th><th>Ograničenja tradicionalnih pristupa</th><th colspan="3">Nova metodika</th></tr>
<tr><td rowspan="2">Analizirati poslovni proces i mapirati njegovo izvođenje</td><td rowspan="2">Tradicionalno modeliranje ne uključuje svaki puta provjeru ispravnosti modeliranog poslovnog procesa u simuliranom okruženju prije implementacije.</td><td colspan="3">Analizirati proizvodni poslovni proces i evidentirati sve aktivnosti, odnosno, mapirati ga od početka do kraja izvođenja.</td></tr>
<tr><td colspan="3">Izraditi procesnu simulaciju kako bi se omogućila usporedba njezinih rezultata s izvođenjem proizvodnog digitalnog blizanca te utvrdila eventualna odstupanja.</td></tr>
<tr><td rowspan="3">Izraditi model poslovnog procesa u njegovom trenutnom stanju</td><td rowspan="3">Ne uključuje unaprijed definirane mehanizme za automatizirano prilagođavanje modela stvarnim podacima iz proizvodnje.</td><td rowspan="3">Razviti digitalnog blizanca proizvodnog poslovnog procesa s obzirom na njegovo stanje.</td><td colspan="2">Ugraditi složenije poslovno pravilo, element umjetne inteligencije, u model digitalnog blizanca.</td></tr>
<tr><td>Ugraditi element za komunikaciju digitalnog blizanca proizvodnog poslovnog procesa s vanjskim sustavom.</td><td>Izraditi proizvodnog digitalnog blizanca u alatu za izradu procesno servisnih arhitektura.</td></tr>
<tr><td></td><td>Izraditi Python skriptu za komunikaciju vanjskog sustava s modelom digitalnog blizanca.</td></tr>
<tr><td>Pustiti model poslovnog procesa u pogon</td><td>Nema.</td><td colspan="3">Pustiti digitalnog blizanca proizvodnog poslovnog procesa u pogon.</td></tr>
<tr><td rowspan="2">Pratiti uspješnost izvedbe modela poslovnog procesa</td><td rowspan="2">Tradicionalni pristup ne koristi stvarne podatke iz proizvodnog sustava za validaciju modela.</td><td colspan="3">Pratiti izvođenje proizvodnog digitalnog blizanca temeljem stvarnih podataka prikupljenih iz vanjskog sustava.</td></tr>
<tr><td colspan="3">Usporediti rezultate simuliranog modela s modelom digitalnog blizanca.</td></tr>
<tr><td>Optimizirati model poslovnog procesa</td><td>Optimizacija se temelji na statičkim analizama i povijesnim podacima, umjesto kontinuiranog nadzora i adaptacije u stvarnom vremenu.</td><td colspan="3">Optimizirati rad digitalnog blizanca proizvodnog poslovnog procesa uklanjanjem identificiranih neučinkovitosti.</td></tr>
</table>

Tablica 2 Usporedba tradicionalnih pristupa modeliranja poslovnih procesa i nove metodike

12 Primjena umjetne inteligencije u razvoju digitalnih blizanaca

Umjetna inteligencija (eng., *Artificial Intelligence* (AI)) i digitalni blizanci se mogu zajedno koristiti. Umjetna inteligencija može strukturirati ulaze i sintetizirati izlaze digitalnih blizanaca, a digitalni blizanci mogu pružiti robusnu okolinu za testiranje i učenje za AI. Korištenjem ovih dviju tehnologija, organizacije mogu postići eksponencijalno više nego što bi mogle korištenjem samo jedne od navedenih tehnologija (McKinsey&Company, 2024).

Postoji nekoliko klasifikacija umjetne inteligencije, ali za potrebe ovog istraživanja spomenimo kako osnovne vrste umjetne inteligencije uključuju usku (ili slabu) umjetnu inteligenciju, koja je specijalizirana za određene zadatke, poput prepoznavanja slika ili analize podataka, i opću (ili jaku) umjetnu inteligenciju, koja može razumjeti, učiti i rješavati probleme na razini ljudske inteligencije.

Ako je digitalni blizanac pokretan kombinacijom umjetne inteligencije, poput rudarenja podataka primjenom klaster analize i tehnika odlučivanja (Križanić, S., 2020), te dubokog učenja, tada može vjerno odražavati svog fizičkog blizanca i omogućiti pravovremeno otkrivanje potencijalnih problema (Dymitrowski, A. & Mielcarek, P, 2021), oslanjajući se na niz senzora ugrađenih u fizički svijet za prijenos podataka u realnom vremenu o operativnom procesu i okruženju (Marmolejo-Saucedo, J.A., 2020), (Romero, D. i ostali, 2021), (Dashkina, A. i ostali, 2020). Podaci prikupljeni s povezanih senzora se zatim analiziraju u oblaku i dostupni su putem kontrolne ploče alata u kojem se izgradio digitalni blizanac. U studiji slučaja provedenoj u radu (Marmolejo-Saucedo, J.A., 2020) su u model digitalnog blizanca ugradili strojno učenje i algoritme za prepoznavanje uzoraka koji mogu pomoći u identifikaciji promjenjivih trendova u opskrbnom lancu i potražnji. Autori u radovima (Dashkina, A. i ostali, 2020), (Hielscher, T. i ostali, 2023) navode kako su se neuronske mreže pokazale učinkovitima u rješavanju kompleksnih problema vezanih uz obradu podataka te ih preporučuju za izradu digitalnih blizanaca. Cilj primjene neuronskih mreža u izradi digitalnog blizanca poslovnog procesa je njegova optimizacija i realnije modeliranje stvarnih poslovnih sustava (Beke, Á.K. i ostali, 2021). Neuronske mreže se mogu prilagoditi različitim poslovnim potrebama i smatraju se korisnima za automatizaciju, optimizaciju i poboljšanje upravljanja poslovnim procesima. Ako se model poslovnog procesa ne može opisati na jednostavan način, npr., diferencijalnim jednadžbama, preporučuje se korištenje neuronskih mreža za izradu digitalnih blizanaca (Dashkina, A. i ostali, 2020). Usporedba učinaka metoda strojnog učenja u radu (Hielscher, T. i ostali, 2023) je rezultirala zaključkom kako su neuronske mreže prikladne za korištenje u izradi digitalnog blizanca. U radu (Beke, Á.K. i ostali, 2021) je opisan razvoj digitalnog blizanca u farmaceutskoj industriji koji uključuje neuronske mreže, eksperimentalne skupove podataka korištene za treniranje modela ili prilagodbu (eng., *fitting*) i validaciju modela digitalnog blizanca. Razvojem digitalnog blizanca poslovnog procesa, dobiva se automatizacija procesa čime se postiže modernizacija farmaceutske industrije. Vaskovsky i dr. (2020) diskutiraju o izradi digitalnog blizanca i korištenju tehnologije neuronskih mreža za personalizirane prehrambene proizvode za ljude s genetskom sklonošću dijabetesu. Park i Van Der Aalst (2021) predlažu model digitalnog blizanca kao prikaz stvarnog poslovnog procesa i podršku informacijskom sustavu organizacije.

Neuronska mreža je masovno paralelni distribuirani procesor sastavljen od jednostavnih procesnih jedinica, koji ima prirodnu sklonost pohranjivanja iskustvenog znanja i njegovoj dostupnosti za upotrebu. Podsjeća na ljudski mozak po dva aspekta: 1. mreža stječe znanje iz svog okruženja kroz proces učenja. 2. Snage povezanosti među neuronima, poznate kao sinaptičke težine, koriste se za pohranu stečenog znanja (Simon Haykin, 2009). Neuronske mreže mogu otkriti statističke pravilnosti i prilagođavati svoje parametre čak i u promjenjivim okruženjima (Raul Rojas, 1996).

Osim izbora odgovarajuće metode umjetne inteligencije, u razvoju digitalnog blizanca važno je razmotriti i računalne zahtjeve pojedinih algoritama te mogućnosti njihove izvedbe u kontekstu stvarnog industrijskog okruženja, primjerice na *edge* uređajima. Jednostavniji algoritmi strojnog učenja, poput stabla odlučivanja ili regresije, imaju nižu računalnu složenost i pogodni su za implementaciju izravno na *edge*-u, omogućujući brzu lokalnu obradu i nisku latenciju. S druge strane, kompleksniji algoritmi poput neuronskih mreža zahtijevaju više računalnih resursa, osobito tijekom faze treniranja. U ovom istraživanju treniranje neuronske mreže provedeno je s ciljem izgradnje prediktivnog modela za potrebe digitalnog blizanca proizvodnog poslovnog procesa. Iako u konkretnom slučaju *edge* obrada nije korištena, tehnički je moguće digitalnog blizanca realizirati na više načina, ovisno o složenosti problema i zahtjevima sustava. U industrijskoj praksi, ako je nužno donošenje odluka u stvarnom vremenu, primjerice kod detekcije kvarova, dijelovi obrade (ili pojednostavljeni modeli) mogu se, primjerice, premjestiti na *edge* uređaje. Tako *edge* može biti funkcionalna komponenta digitalnog blizanca, posebno za zadatke koji zahtijevaju brzu reakciju, dok se složeniji modeli mogu razvijati i održavati lokalno ili u distribuiranom okruženju. Takav fleksibilan pristup omogućuje prilagodbu arhitekture digitalnog blizanca stvarnim tehničkim i vremenskim ograničenjima, pri čemu se ravnoteža postiže između efikasnosti obrade, zahtjeva sustava i dostupnih resursa. Odabir neuronskih mreža u ovom radu temelji se na njihovoj dokazanoj sposobnosti prepoznavanja nelinearnih obrazaca i primjenjivosti na kompleksne procese, uz svijest o mogućnostima pojednostavljenja modela za buduće izvedbe na *edge*-u.

U ovom istraživanju odabrane su neuronske mreže kao metoda umjetne inteligencije za proširenje funkcionalnosti proizvodnog digitalnog blizanca, budući da su se u dosadašnjim istraživanjima pokazale kao učinkovito rješenje za složene zadatke razvoja digitalnih blizanaca, osobito kada je riječ o kompleksnijim poslovnim procesima i obradi velike količine podataka (Dashkina, A. i ostali, 2020), (Hielscher, T. i ostali, 2023), (Beke, Á.K. i ostali, 2021). Iako postoje jednostavniji algoritmi poput stabla odlučivanja, koji se također mogu koristiti u specifičnim i jednostavnijim slučajevima, neuronske mreže omogućuju veću fleksibilnost i sposobnost prilagodbe promjenjivim uvjetima te otkrivanje nelinearnih obrazaca u podacima. Upravo zbog njihove učestalosti u relevantnoj literaturi i sposobnosti da modeliraju jednostavne i kompleksnije sustave, neuronske mreže su u ovom istraživanju odabrane kao metoda za izradu prediktivnog modela digitalnog blizanca proizvodnog poslovnog procesa.

12.1 Uloga digitalnog blizanca u prediktivnom održavanju

Predviđanje održavanja strojeva, koji se koriste u proizvodnji, u kombinaciji s digitalnim blizancima proizvodnih poslovnih procesa se može koristiti radi razvoja modela koji prikazuju stvarno stanje i ponašanje strojeva i druge proizvodne opreme u stvarnom vremenu. Na taj način model digitalnog blizanca proizvodnog poslovnog procesa omogućava stvarni prikaz stvarnih podataka o strojevima i

proizvodnoj opremi, uključujući senzorske podatke (poput temperature, vibracija ili tlaka) i radne parametre. Kada se prediktivno održavanje integrira u model digitalnog blizanca, moguće je:

- uspostaviti dinamički model koji stalno prati rad opreme i ažurira predviđanja o potencijalnim problemima i kvarovima u stvarnom vremenu,
- simulirati scenarije u kojima se simuliraju različiti uvjeti rada, a to pomaže u testiranju scenarija za održavanje i usporedbi različitih „što-ako“ scenarija,
- optimizirati resurse i troškove planiranjem unaprijed i pravovremenim raspoređivanjem resursa na temelju predviđanja.

Tako su prednosti prediktivnog održavanja kod proizvodnog digitalnog blizanca: povećana dostupnost opreme, smanjenje neplaniranih zastoja jer digitalni blizanac upozorava na probleme i kvarove prije nego što se oni dogode, smanjenje troškova proizvodnje jer je bolja raspodjela resursa temeljem realnih podataka i predikcija, brža reakcija na promjene jer digitalni blizanac reagira na promjene u podacima u stvarnom vremenu.

Cilj korištenja umjetne inteligencije u proizvodnom digitalnom blizancu je bolje predviđanje događaja koji se mogu pojaviti tijekom izvođenja poslovnog procesa i koji mogu utjecati na ishod izvođenja procesa. Razlozi za uključivanje elementa umjetne inteligencije u razvoj digitalnih blizanaca proizvodnih poslovnih procesa mogu biti:

- optimizacija procesa i resursa,
- prediktivno održavanje,
- unaprjeđenje donošenja operativnih odluka,
- unaprjeđenje kvalitete razvoja proizvoda,
- smanjenje troškova razvoja proizvoda,
- smanjenje troškova i vremena testiranja.

Algoritmi umjetne inteligencije u razvoju digitalnih blizanaca omogućavaju da se analiziraju velike količine podataka i da se uoče skriveni uzorci u podacima, što može pomoći u optimizaciji proizvodnih resursa, kao što su proizvodni materijali, energija, ljudski rad i strojevi. Dodatno, uz pomoć strojnog učenja, digitalni blizanac proizvodnog poslovnog procesa može predvidjeti kvarove i optimalno vrijeme za održavanje opreme i uređaja, čime se smanjuju neočekivani zastoji te se produžuje vijek trajanja strojeva i opreme. To je vrlo važno kako bi se osigurao kontinuitet proizvodnje. Nadalje, umjetna inteligencija može poboljšati donošenje operativnih odluka u stvarnom vremenu na temelju predviđanja. Digitalni blizanci s ugrađenim mehanizmom umjetne inteligencije mogu testirati različite scenarije i preporučiti najbolje akcije kako bi se postignuli poslovni ciljevi, kao što je smanjenje troškova ili povećanje produktivnost.

U brojnim recentnim istraživanjima primjena umjetne inteligencije u digitalnim blizancima temelji se na korištenju algoritama strojnog učenja za analizu podataka u stvarnom vremenu i za predviđanje stanja fizičkog sustava. Takav pristup istaknut je, primjerice, u radovima (Marmolejo-Saucedo, J.A., 2020) i (Dashkina, A. i ostali, 2020), gdje su modeli strojnog učenja ugrađeni radi poboljšanja analize podataka iz senzora i optimizacije opskrbnih lanaca. Međutim, navedena istraživanja najčešće obrađuju tehnički aspekt integracije umjetne inteligencije, bez preciznog pozicioniranja unutar metodologije razvoja digitalnog blizanca. U ovom istraživanju potrebno je razviti integrirani pristup u kojem će komponenta umjetne inteligencije, točnije, neuronska mreža za prediktivno održavanje,

biti ugrađena kao sastavni dio šireg metodološkog okvira razvoja proizvodnog digitalnog blizanca. Element umjetne inteligencije je potrebno implementirati u obliku složenijeg poslovnog pravila unutar digitalnog blizanca, pri čemu je potrebno najprije razviti i istrenirati prediktivni model za održavanje, a zatim isti integrirati u model poslovnog procesa putem mehanizma za povezivanje s automatiziranim aktivnostima. Na taj način komponenta umjetne inteligencije ne bi djelovala izolirano već bi bila kontekstualizirana unutar procesa, čime bi se osigurala usklađenost modela s logikom proizvodnog procesa. Novina predloženog pristupa ovog istraživanja ogleda se u metodološkom povezivanju prediktivnog modela temeljenog na neuronskim mrežama s BPMN modelom poslovnog procesa, što može omogućiti automatiziranu i adaptivnu reakciju digitalnog blizanca na promjene u realnom vremenu, temeljem učenja iz podataka. Na ovaj način se može doprinijeti integraciji umjetne inteligencije ne samo na razini analize podataka, nego i na razini upravljanja i optimizacije samog izvođenja poslovnog procesa.

12.2 Podaci korišteni za slučajeve prediktivnog održavanja strojeva

Za potrebe izrade ovog istraživanja korištena je kombinacija podataka prikupljenih u razmatranim studijama slučajeva i Kaggleov skup podataka za prediktivno održavanje dostupan javno sa stranice: https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification.
Ukupan broj zapisa preuzetih s Kaggle-a iznosi 10.000 i oni su korišteni isključivo za treniranje neuronskih mreža u svrhu izgradnje prediktivnog modela održavanja. S druge strane, stvarni podaci iz dvaju konkretnih slučajeva korištenja predstavljaju temelj za razvoj, validaciju i evaluaciju proizvodnih digitalnih blizanaca, jer sadržavaju kontekstualne informacije o stvarnom izvođenju poslovnih procesa, vremenskim tokovima, korištenim resursima i redoslijedu aktivnosti. Podaci s Kaggle-a su korišteni u ograničenoj ulozi, a to je kao dopunski izvor za izgradnju prediktivnog modela. Ključni elementi istraživanja, uključujući modeliranje procesa, izvođenje digitalnih blizanaca i analiza ponašanja sustava, temelje se primarno na stvarnim podacima, koji zbog svoje autentičnosti i povezanosti s konkretnim procesima nose znatno veću informacijsku vrijednost. Bitno je za istaknuti kako su podaci prikupljani iz stvarnih slučajeva dinamične prirode, odnosno njihov broj kontinuirano raste s vremenom kako se odvija proizvodni proces i kako sustav prikuplja nove informacije o izvršavanju poslovnih aktivnosti. Nasuprot tome, Kaggle podaci su statični i zatvoreni skup. Kako bi se koncept umjetne inteligencije uključio u razvoj digitalnih blizanaca proizvodnih poslovnih procesa, izrađeno je i trenirano nekoliko klasifikacijskih neuronskih mreža.

U okviru izrade digitalnih blizanaca proizvodnih poslovnih procesa, implementiranih u alatu Camunda, predviđeno je integrirati rezultate prediktivnog održavanja opreme. Rezultati prediktivnog održavanja temelje se na modelima klasifikacijskih neuronskih mreža razvijenih u programskom jeziku Python. Za potrebe razvoja i treniranja neuronskih mreža korišteni su podaci o radu strojeva. Ovaj pristup omogućava analizu različitih scenarija održavanja opreme unutar proizvodnog digitalnog blizanca, čime se doprinosi poboljšanju prediktivnih kapaciteta i optimizaciji proizvodnih poslovnih procesa.

Koraci izrade su:

1. Prikupiti i pripremiti podatke: Uvesti povijesne podatke o kvarovima strojeva i zastojima u proizvodnji iz Kaggle skupa podataka te ih pripremiti za analizu i modeliranje.
2. Razviti prediktivne modele: Izraditi neuronske mreže u Pythonu koje će, na temelju povijesnih podataka o stanju strojeva, predviđati potencijalne zastoje u proizvodnji i kvarove na opremi. Proces uključuje definiranje arhitekture mreže, treniranje, validaciju i optimizaciju modela.
3. Implementirati rezultate prediktivnih modela: Uvesti rezultate prediktivnog održavanja unutar proizvodnog digitalnog blizanca razvijenog u Camundi. Ovi rezultati omogućit će prilagodbu izvođenja proizvodnog poslovnog procesa u stvarnom vremenu, temeljem predviđanja o stanju opreme i potencijalnim kvarovima.

Implementacija koraka izrade je detaljnije razrađena u nastavku.

12.2.1 Prikupljanje i priprema podataka

U ovom će se dijelu opisati podaci korišteni za provedbu istraživanja. Naime, dio podataka je stvarni skup podataka iz razmatranih studija slučajeva, a dio podataka je javni skup podataka objavljen na stranici Kaggle (Kaggle, 2020) budući da je stvarne skupove podataka za prediktivno održavanje opreme općenito teško pronaći kako bi bila dovoljno velika količina podataka za treniranje neuronske mreže, a nije ih jednostavno ni objaviti. Osim toga, kvarovi na opremi se ni ne događaju često. Ovaj skup podataka je korišten za oba slučaja korištenja, i kod razvoja prediktivnog modela održavanja 3D printera i kod razvoja prediktivnog modela održavanja bušilice i preše. Za treniranje neuronske mreže primijenjen je pristup transfer učenja (eng., *transfer learning*), pri čemu je model inicijalno treniran na javno dostupnom Kaggle skupu podataka koji sadrži karakteristike relevantne za industrijsko prediktivno održavanje. Takav pristup omogućio je prijenos naučenih obrazaca iz srodne domene na konkretne slučajeve korištenja u ovom istraživanju, unatoč razlikama u opremi. Skup podataka s Kagglea, korišten u ovom istraživanju, je sintetički skup podataka koji odražava stvarne situacije prediktivnog održavanja u industriji. Aditivna proizvodnja, poznatija kao 3D ispis, predstavlja jedan od ključnih razvojnih smjerova suvremene industrije. Inovacijski park Biel uvelike se oslanja na tehnologiju 3D ispisa kao temeljnu komponentu svojih inovativnih procesa. Slično tome, prikazana studija slučaja usko je povezana s ovom tehnologijom, čime se demonstrira njezina praktična primjena. Stoga podatke o održavanju vežemo uz 3D ispis jer pružaju izvrstan temelj za ilustraciju uvida i značaja ove studije slučaja.
Skup podataka se sastoji od 10.000 slogova pažljivo odabranih da reflektiraju stanje proizvodnje u razmatranim studijama slučaja i pohranjenih u redovima s 13 obilježja prikazanih u stupcima. U skupu podataka nema nedostajućih podataka. Obilježja koja opisuju podatke su (Kaggle, 2020):

- *UID:* jedinstveni identifikator u rasponu od 1 do 10.000
- *productID*: ID proizvoda koji se sastoji od slova L, M ili H za nisku (50% svih proizvoda), srednju (30%) i visoku (20%) varijantu kvalitete proizvoda te serijskog broja specifičnog za varijantu
- *air temperature*: temperatura zraka generirana pomoću procesa slučajne šetnje, a kasnije normalizirana na standardnu devijaciju od 2 K oko 300 K
- *process temperature [K]*: temperatura procesa generirana pomoću procesa slučajne šetnje, normalizirana na standardnu devijaciju od 1 K, dodana na temperaturu zraka uvećanu za 10 K

- *torque [Nm]*: moment zakretanja su vrijednosti normalno raspodijeljene oko 40 Nm s σ = 10 Nm i bez negativnih vrijednosti
- *tool wear [min]*: trošenje alata gdje varijante kvalitete H/M/L dodaju 5/3/2 minute trošenja alata na korišteni alat u procesu, te obilježje „kvar stroja“ koje označava da li je stroj bio neispravan u navedenom slogu zapisa, ako je bilo koji od načina kvara istinit.

Dodatno, u skupu podataka postoje dvije ciljane varijable:

- *target:* kvar ili bez kvara
- *failure type:* označava tip kvara

Br. varijable	**Ime varijable**	**Uloga**	**Tip podatka**	**Jedinica**
1.	*UID*	ID	Cjelobrojni	-
2.	*Produkt ID*	ID	Kategorijski	-
3.	*Tip*	Obilježje	Kategorijski	-
4.	*Temperatura zraka*	Obilježje	Kontinuirani	K
5.	*Temperatura procesa*	Obilježje	Kontinuirani	K
6.	*Moment zakretanja*	Obilježje	Kontinuirani	Nm
7.	*Trošenje alata*	Obilježje	Cjelobrojni	Min
8.	*Kvar stroja*	Ciljana varijabla	Cjelobrojni	-
9.	*Kvar uslijed trošenja alata*	Ciljana varijabla	Cjelobrojni	-
10.	*Kvar uslijed disipacije topline*	Ciljana varijabla	Cjelobrojni	-
11.	*Kvar uslijed napajanja*	Ciljana varijabla	Cjelobrojni	-
12.	*Kvar uslijed preopterećenja*	Ciljana varijabla	Cjelobrojni	-
13.	*Nasumični kvarovi*	Ciljana varijabla	Cjelobrojni	-

*Tablica 3 Podaci o varijablama (*AI4I 2020 Predictive Maintenance Dataset, *2020)*

Kao što se može vidjeti iz tablice 3, postoji pet vrsta kvarova stroja (*AI4I 2020 Predictive Maintenance Dataset*, 2020):

1. Kvar uslijed trošenja alata: alat će biti zamijenjen ili će doći do kvara u nasumično odabranom vremenu trošenja alata između 200 i 240 minuta (120 puta u priloženom skupu podataka). U tom trenutku, alat je zamijenjen 69 puta, a došlo je do kvara 51 put (nasumično dodijeljeno).
2. Kvar uslijed disipacije topline: disipacija topline uzrokuje kvar procesa ako je razlika između temperature zraka i temperature procesa manja od 8,6 K. Takvih je 115 zapisa.
3. Kvar uslijed napajanja: proizvod momenta jednak je snazi potrebnoj za proces. Ako je ta snaga manja od 3500 W ili veća od 9000 W, poslovni proces ne uspijeva izvođenje, što se događa 95 puta u priloženom skupu podataka.
4. Kvar uslijed preopterećenja: ako je proizvod trošenja alata i momenta veći od 11.000 minNm za L varijantu proizvoda (12.000 M, 13.000 H), poslovni proces ne uspijeva izvođenje zbog preopterećenja. Takvih je 98 zapisa.
5. Nasumični kvarovi: svaki poslovni proces ima 0,1 % šanse da ne uspije, neovisno o parametrima procesa. Takvih je samo 5 slučajeva, što je manje od očekivanog za 10.000 zapisa u priloženom skupu.

Primjer kako izgleda skup podataka za treniranje neuronskih mreža je prikazan na slici 17 u nastavku.

	A	B	C	D	E	F	G	H	I
1	UID	Product ID	Type	Air temperature [K]	Process temperature [K]	Torque [Nm]	Tool wear [min]	Target	Failure Type
2	1	M14860	M	298.1	308.6	42.8	0	0	No Failure
3	2	L47181	L	298.2	308.7	46.3	3	0	No Failure
4	3	L47182	L	298.1	308.5	49.4	5	0	No Failure
5	4	L47183	L	298.2	308.6	39.5	7	0	No Failure
6	5	L47184	L	298.2	308.7	40	9	0	No Failure
7	6	M14865	M	298.1	308.6	41.9	11	0	No Failure
8	7	L47186	L	298.1	308.6	42.4	14	0	No Failure
9	8	L47187	L	298.1	308.6	40.2	16	0	No Failure
10	9	M14868	M	298.3	308.7	28.6	18	0	No Failure
11	10	M14869	M	298.5	309	28	21	0	No Failure
12	11	H29424	H	298.4	308.9	23.9	24	0	No Failure
13	12	H29425	H	298.6	309.1	44.3	29	0	No Failure
14	13	M14872	M	298.6	309.1	51.1	34	0	No Failure
15	14	M14873	M	298.6	309.2	30	37	0	No Failure
16	15	L47194	L	298.6	309.2	19.6	40	0	No Failure
17	16	L47195	L	298.6	309.2	48.4	42	0	No Failure
18	17	M14876	M	298.6	309.2	46.6	44	0	No Failure
19	18	M14877	M	298.7	309.2	45.6	47	0	No Failure
20	19	H29432	H	298.8	309.2	54.5	50	0	No Failure
21	20	M14879	M	298.9	309.3	32.5	55	0	No Failure
22	21	H29434	H	298.9	309.3	42.7	58	0	No Failure
23	22	L47201	L	298.8	309.3	44.8	63	0	No Failure
24	23	M14882	M	298.9	309.3	30.7	65	0	No Failure
25	24	L47203	L	299	309.4	25.7	68	0	No Failure
26	25	M14884	M	299	309.4	37.3	70	0	No Failure

Slika 17 Prikaz dijela skupa podataka za treniranje neuronskih mreža

Razjasnimo zašto je ovo dobar skup podataka za prikaz uvjeta pod kojima rade 3D printeri kako bi se omogućilo prediktivno održavanje strojeva u proizvodnom digitalnom blizancu. Temperatura zraka je važna kod 3D printera zbog procesa hlađenja, posebno kod printera koji koriste vlakno ili za stvrdnjavanje smole. Temperatura procesa je ključna za proces ispisa jer uključuje temperaturu mlaznice (FDM (eng., *Fused Deposition Modeling*)) ili temperaturu stvrdnjavanja (SLA (stereolitografija) / DLP (eng., *Digital Light Processing*)). Temperatura procesa je ključna u 3D ispisu jer izravno utječe na ponašanje materijala tijekom ispisa, osiguravajući pravilno prianjanje slojeva, strukturni integritet i ukupnu kvalitetu ispisa. Kod FDM printera filament se mora zagrijati na određenu temperaturu kako bi se otopio i protjecao kroz mlaznicu. Ako je temperatura preniska, filament se možda neće pravilno istiskivati, što dovodi do slabih ili nepotpunih ispisa. Ako je previsoka, materijal može degradirati, uzrokujući lošu kvalitetu ispisa ili začepljenje mlaznice. Nadalje, taljeni filament mora zadržati odgovarajuću temperaturu kako bi se vezao s prethodno ispisanim slojem. Nepravilne temperature mlaznice mogu rezultirati lošim prianjanjem slojeva, uzrokujući odvajanje ili deformaciju ispisa. Kod SLA ili DLP printera, temperatura stvrdnjavanja igra ulogu u kemijskoj reakciji koja učvršćuje tekuću smolu. Kontrolirano stvrdnjavanje osigurava da ispisani objekt postigne željenu čvrstoću i preciznost. Nepravilno stvrdnjavanje može dovesti do nedovoljno učvršćenih (mekih ili nepotpunih) dijelova ili prekomjernog učvršćivanja (gubitka detalja i krhkosti). Regulacija temperature pomaže u održavanju potrebne konzistencije procesa stvrdnjavanja. Moment zakretanja je relevantan za koračne motore koji pokreću osi ili istiskivače. Preveliki moment mogao bi ukazivati na opterećenje motora. Trošenje alata je bitan element zbog praćenja istrošenosti mlaznice (kod FDM printera) ili drugih potrošnih dijelova poput oštrica za ponovno premazivanje kod SLS pisača.

Nadalje, obrazložimo i zašto je ovaj skup podataka dobar za prikaz uvjeta pod kojima rade bušilica i preša koje se koriste u pametnoj tvornici na FSB-u. Bušilice i preše su važne u proizvodnji mobilnih uređaja, osobito za montažu komponenti, precizno bušenje i oblikovanje metalnih ili plastičnih dijelova. Ključni parametri, poput temperature zraka, temperature procesa, momenta zakretanja i istrošenost alata, ključni su za praćenje rada bušilica i preša te pomažu u otkrivanju problema kao što su pregrijavanje, mehaničko trošenje ili degradacija alata. Temperatura zraka i procesna temperatura

su korisne za prepoznavanje problema s pregrijavanjem u operacijama preciznog bušenja. Moment zakretanja je relevantan za otkrivanje mehaničkog trošenja ili neusklađenosti vretena ili alata. Istrošenost alata je izravna mjera stanja alata, koja omogućuje pravovremeno planiranje preventivnog održavanja i sprječavanje zastoja. Ovi parametri odražavaju stvarne scenarije u proizvodnim procesima visoke preciznosti, gdje je održavanje optimalnog rada strojeva presudno za kontrolu kvalitete. Analizom ovih parametara prediktivni model može identificirati obrasce povezane s nadolazećim kvarovima stroja, što omogućuje pravovremene intervencije. Klasifikacija kvarova u skupu podataka, koja obuhvaća vrste kvarova poput istrošenosti alata, problema s odvođenjem topline i preopterećenja, vjerodostojno prikazuje izazove u održavanju bušilica i preša koji se inače mogu dogoditi. Primjerice, istrošenost alata bušilice uzrokuje postupno pogoršanje kvalitete bušenja i povećava količinu otpada, dok problem s odvođenjem topline u brzim operacijama bušenja može uzrokovati oštećenja alata i obratka. Preopterećenje stroja može dovesti do oštećenja mehaničkih komponenti. Ovi kvarovi ističu važnost prediktivnog održavanja bušilica i preša, koje može smanjiti neplanirane zastoje i poboljšati ukupnu učinkovitost opreme.
Iako Kaggle skup podataka nije prikupljen izravno na korištenim strojevima, njegovi parametri pokazuju značajnu kompatibilnost s našim slučajevima korištenja. Kod 3D printera, temperatura procesa odgovara temperaturi hotenda i ležaja, moment zakretanja ekstrudera i osi, te trošenje komponenti poput mlaznica. Kod bušilica i preša, temperature procesa reflektiraju zagrijavanje tijekom obrade, moment zakretanja odgovara opterećenju alata, dok trošenje alata direktno korelira s životnim ciklusom bušilice i preše. Tipovi kvarova iz skupa, poput trošenja alata, preopterećenja i disipacije topline, manifestiraju se i na opremi koja se upotrebljava u slučajevima korištenja: zamjena istrošenih mlaznica kod 3D printera, preopterećenje motora kod bušilice, te pregrijavanje elektronike.

12.2.2 Razvoj prediktivnog modela

Kako bi se razvili prediktivni modeli o održavanju strojeva, a to su u ovom istraživanju 3D printeri, bušilica i preša, napravit će se nekoliko Python skripti u funkciji *utility* servisa u kojima će biti razvijene neuronske mreže. Npr., Python skripta za održavanje 3D printera će služiti za razvoj i treniranje dva modela neuronskih mreža za prediktivno održavanje opreme koristeći podatke o proizvodnom procesu, te će omogućiti predviđanje kvarova. Radit će na principu da se podaci učitavaju iz CSV datoteke, pri čemu se određeni atributi (npr. „*Product ID*“) pretvaraju u kategorijske varijable. Numerička obilježja (npr. temperatura) se standardiziraju kako bi se osigurala uspješna obrada u neuronskoj mreži. Podatkovni skup je podijeljen na ulazna obilježja i ciljne varijable. Kao i kod digitalnog blizanca proizvodnje dronova, gdje će prediktivna analitika imati ulogu pratiti stanje printera, tako će i kod proizvodnje mobilnih uređaja biti korištena neuronska mreža kako bi se omogućilo prediktivno održavanje bušilice i preše. Za tu svrhu napravit će se Python skripta koja će imati dvostruku funkciju, a služit će za treniranje neuronske mreže koristeći podatke iz Access baze podataka o uvjetima rada proizvodne opreme te predviđanje kvara i vrste kvara nad opremom na temelju novih podataka ili nasumično odabranog zapisa iz baze, putem REST API-ja izgrađenog pomoću Flask aplikacije.

12.2.3 Implementacija prediktivnih modela u proizvodne digitalne blizance

U poglavlju *Koraci razvoja digitalnog blizanca proizvodnog poslovnog procesa* su prezentirani koraci razvoja digitalnog blizanca proizvodnog poslovnog procesa. U koraku modeliranja se navodi

kako je potrebno „*Ugraditi složenije poslovno pravilo, element umjetne inteligencije, u model digitalnog blizanca*“. U suštini se ovaj korak sastoji od nekoliko podkoraka koji uključuju izradu prediktivnog modela koji bi se uključio u digitalnog blizanca proizvodnog poslovnog procesa.

Koraci izgradnje prediktivnog modela i njegovog uključivanja u proizvodni digitalni blizanac su tako:

1. Pribaviti skup podataka o održavanju opreme.
2. Izraditi prediktivni model za predviđanje održavanja strojeva: Razviti klasifikacijsku neuronsku mrežu s nekoliko slojeva za prepoznavanje obrazaca u vremenskim serijama.
3. Trenirati neuronsku mrežu na skupu za treniranje kako bi prepoznala složene obrasce i izvanredne događaje koji mogu uzrokovati kvarove i padove te ju validirati na validacijskom skupu.
4. Ugraditi prediktivni model održavanja u proizvodni digitalni blizanac
    a. Spremiti model i implementirati ga unutar proizvodnog digitalnog blizanca za izračunavanje predikcija.
    b. Povezati alat za izradu procesno servisnih aplikacija (npr., Camunda Modeler) s vanjskim sustavom kako bi se podaci prikupljali u stvarnom vremenu i obrađivali kroz model digitalnog blizanca.
5. Periodički optimizirati prediktivni model
    a. Periodički optimizirati prediktivni model temeljem stvarnih podataka kako bi se prilagodio promjenama u radu strojeva.
    b. Nadograditi funkcionalnosti digitalnog blizanca na temelju povratnih informacija iz vanjskog sustava i rezultata predikcija.

Implementacija prediktivnih rezultata u proizvodni digitalni blizanac završava integracijom treniranog prediktivnog modela u digitalni blizanac te povezivanjem s vanjskim, odnosno, stvarnim sustavom (preko, npr., IoT uređaja) ako je to moguće, za kontinuirano prikupljanje podataka i izračun predikcija u stvarnom vremenu. Time se omogućuje pravovremeno prepoznavanje potencijalnih kvarova i optimizacija održavanja strojeva, čime se smanjuju neplanirani zastoji i povećava operativna učinkovitost.

13 Metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa

Razvoj metodike u ovom istraživanju temelji se na sintezi razvojnih pristupa iz triju različitih, ali međusobno povezanih područja: razvoja softverskih rješenja, razvoja simulacijskih modela i inženjeringa sustava automatskog upravljanja. Svako od tih područja ima vlastite razvojne paradigme koje su analizirane kako bi se razumjeli njihovi ključni elementi. U području razvoja softverskih rješenja, temeljna razvojna paradigma je iterativno-inkrementalni pristup, koji podrazumijeva postupnu izgradnju rješenja kroz niz razvojnih ciklusa, pri čemu se svaki ciklus sastoji od faza planiranja, implementacije, testiranja i evaluacije. Ova paradigma omogućuje brzo reagiranje na promjene zahtjeva i prilagodbu funkcionalnosti u skladu s povratnim informacijama. U razmatranje su uzeti principi agilnog razvoja softvera (npr. Scrum), koji naglašavaju fleksibilnost, modularnost i stalnu validaciju funkcionalnosti kroz radne prototipove. U okviru simulacijskog modeliranja, dominantna razvojna paradigma temelji se na procesu definiranja stvarnog sustava kroz formalne modele koji omogućuju eksperimentiranje u kontroliranom okruženju. Ova paradigma uključuje jasno definirane faze: formulaciju problema, izradu konceptualnog modela, formalizaciju u alatu za provođenje simulacija, izvođenje eksperimenta i interpretaciju rezultata. Poseban naglasak stavlja se na replikabilnost i parametarsku prilagodbu modela, kao i mogućnost izvođenja „što-ako“ analiza koje podržavaju donošenje odluka bez narušavanja stvarnog sustava. U području inženjeringa sustava automatskog upravljanja, razvojna paradigma počiva na modelima kontrole zatvorene petlje, gdje se sustavi projektiraju tako da neprekidno nadziru ulazne i izlazne varijable, uz prilagodbu ponašanja na temelju povratnih informacija (eng. *feedback control*). Ova paradigma uključuje strogo definirane faze: analizu postojećih sustava i senzora, specifikaciju upravljačkih zahtjeva, projektiranje arhitekture sustava, te definiranje komunikacijskih protokola i mehanizama nadzora. U obzir se uzimaju i principi industrijske automatizacije kao što su modularnost, odziv u stvarnom vremenu i deterministička pouzdanost. Kombinacija metodoloških elemenata iz ovih triju područja poslužila je kao ulazna osnova za oblikovanje predložene metodike razvoja digitalnog blizanca proizvodnog poslovnog procesa.

Na slici 18 u nastavku se nalazi prikaz metodike razvoja proizvodnog digitalnog blizanca zajedno s koracima ugradnje složenijeg poslovnog pravila te uključivanja komunikacijskog elementa u proizvodni digitalni blizanac. Slika prikazuje metodiku razvoja kroz pet ključnih koraka: analizu, modeliranje, implementaciju, nadzor i optimizaciju. U prvom dijelu, analiza obuhvaća mapiranje svih aktivnosti unutar proizvodnog poslovnog procesa, dok modeliranje uključuje izradu digitalnog blizanca, ugradnju poslovnih pravila temeljenih na umjetnoj inteligenciji te povezivanje modela s vanjskim uređajima putem Python skripti i alata za procesno servisne arhitekture.
Implementacija označava puštanje digitalnog blizanca u pogon, nakon čega slijedi nadzor njegove izvedbe temeljem stvarnih podataka i usporedba simuliranog modela s realnim podacima. Optimizacija se provodi uklanjanjem neučinkovitosti kako bi se poboljšala preciznost i učinkovitost digitalnog blizanca. Paralelno, razvoj prediktivnog modela uključuje prikupljanje podataka o održavanju opreme, izradu i treniranje klasifikacijske neuronske mreže, njezinu integraciju u

proizvodni digitalni blizanac te periodičku optimizaciju kako bi se prilagodila promjenama u radu strojeva.
Cijeli sustav oslanja se na komunikaciju između digitalnog blizanca, prediktivnog modela i vanjskog svijeta, omogućujući periodičko poboljšanje modela i prilagodbu u stvarnom vremenu. Kroz navedenu metodiku, proizvodni digitalni blizanac replicira stvarni, proizvodni, poslovni proces, predviđa potencijalne kvarove te optimizira operacije na temelju analize podataka iz stvarnog svijeta**.**

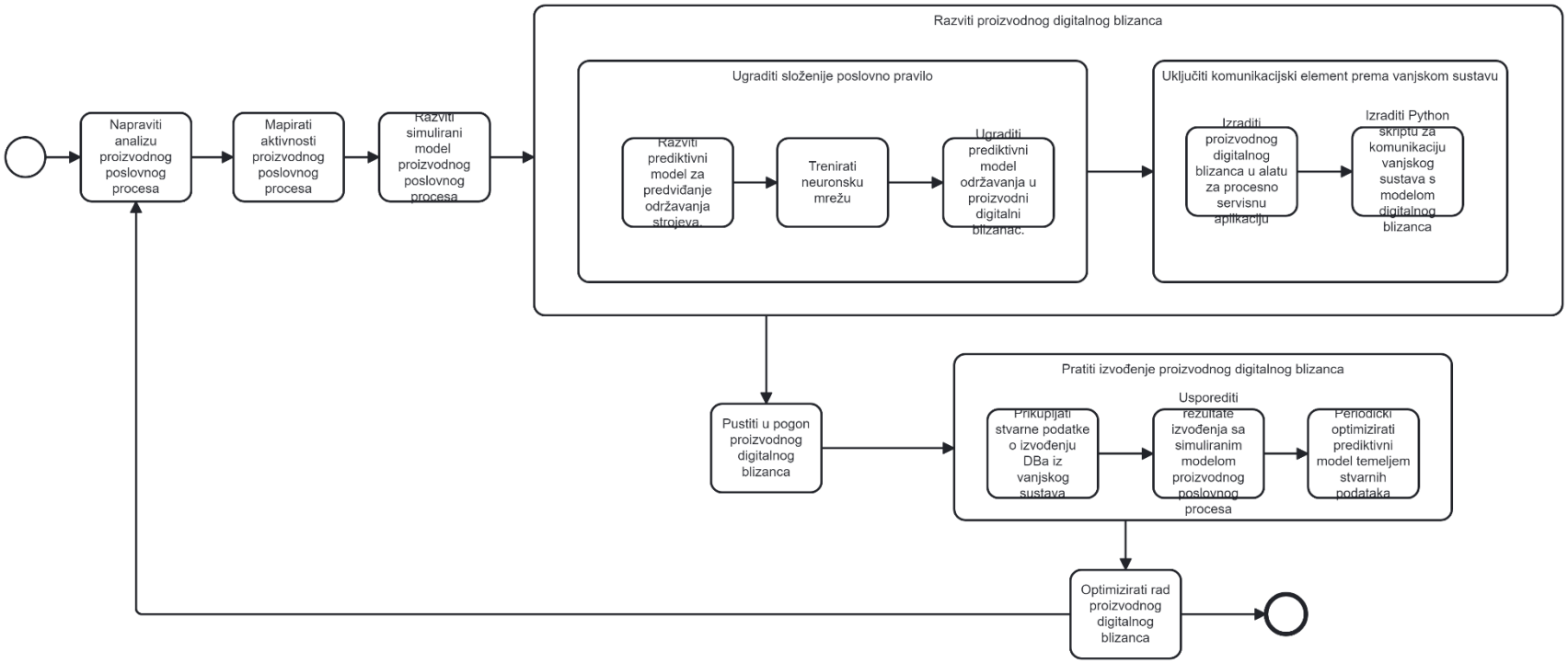
U nastavku se nalazi tablica 4 koja prikazuje aktivnosti metodike razvoja digitalnog blizanca proizvodnog poslovnog procesa uz objašnjenje svake aktivnosti.

Redni br. aktivnosti	**Naziv aktivnosti**	**Opis aktivnosti**
1.	Napraviti analizu proizvodnog poslovnog procesa.	Prikupljanje i proučavanje postojećih podataka, identifikacija ključnih aktivnosti i međusobnih odnosa unutar procesa. Cilj je dobiti uvid u postojeće stanje, prepoznati nedostatke i mogućnosti za optimizaciju.
2.	Mapirati aktivnosti proizvodnog poslovnog procesa.	Vizualizacija i dokumentiranje koraka unutar proizvodnog poslovnog procesa. Poslovni proces se precizno opisuje, uključujući njegove ulazne i izlazne parametre. Ovaj korak osigurava jasno razumijevanje procesa i olakšava njegovu simulaciju i optimizaciju.
3.	Razviti simulirani model proizvodnog poslovnog procesa.	Simulirani model proizvodnog poslovnog procesa konstruira se na temelju prethodne analize i mapiranja. Koristi se računalni alat za modeliranje poput IBM WebSpherea kako bi se stvorila virtualna replika procesa koja omogućuje testiranje različitih scenarija. Model pomaže u predviđanju performansi sustava i identificiranju mogućih poboljšanja.
4.	Razviti proizvodnog digitalnog blizanca.	Integracija simuliranog modela s dinamičkim podacima iz stvarnog proizvodnog sustava. Digitalni blizanac omogućuje praćenje, analizu i optimizaciju poslovnog procesa u realnom vremenu. Ovaj korak obuhvaća povezivanje s relevantnim tehnologijama i postavljanje temelja za napredne funkcionalnosti.
4.1.	Ugraditi složenije poslovno pravilo.	Odnosi se na donošenje odluka unutar digitalnog blizanca. Ugradnja složenijih pravila omogućava bolju automatizaciju procesa, prediktivno održavanje, optimizaciju resursa i prilagodbu modela stvarnim operativnim uvjetima. Ova pravila mogu uključivati prediktivne algoritme i integraciju s vanjskim sustavima.
4.1.1.	Razviti prediktivni model za predviđanje održavanja strojeva.	Prediktivni model temelji se na analizi povijesnih podataka kako bi se unaprijed prepoznali obrasci koji ukazuju na kvarove strojeva. Korištenjem metoda strojnog učenja, model omogućava predviđanje potrebnih intervencija prije nego što dođe do zastoja u proizvodnji. Time se povećava učinkovitost održavanja i smanjuju operativni troškovi.

4.1.2.	Trenirati neuronsku mrežu.	Trening neuronske mreže provodi se nad relevantnim skupovima podataka kako bi model naučio prepoznavati i predviđati kvarove strojeva. Odabiru se odgovarajuće arhitekture mreže, podešavaju hiperparametri i evaluira točnost modela. Cilj je da se postigne visoka pouzdanost predikcija.
4.1.3.	Ugraditi prediktivni model održavanja u proizvodni digitalni blizanac.	Integracija prediktivnog modela u proizvodni digitalni blizanac omogućuje primjenu predikcija u realnom vremenu. Digitalni blizanac koristi izlazne podatke modela za donošenje odluka o optimalnom rasporedu održavanja. Ovaj korak povećava pouzdanost proizvodnog sustava i smanjuje neplanirane zastoje.
4.2.	Uključiti komunikacijski element prema vanjskom sustavu.	Komunikacijski element omogućuje povezivanje digitalnog blizanca s vanjskim, odnosno, stvarnim sustavom iz kojeg se prikupljaju podaci. Implementacija ove aktivnosti omogućuje kontinuirani protok podataka i ažuriranje modela na temelju stvarnih uvjeta rada. Time se poboljšava preciznost predikcija i prilagodljivost blizanca.
4.2.1.	Izraditi proizvodnog digitalnog blizanca u alatu za procesno servisnu aplikaciju.	Razvoj proizvodnog digitalnog blizanca u alatu za razvoj procesno servisnih aplikacija (poput Camunde) omogućuje vizualizaciju i upravljanje proizvodnim procesom. Aplikacija omogućuje korisnicima interakciju s modelom te prilagodbu parametara u skladu s operativnim potrebama.
4.2.2.	Izraditi Python skriptu za komunikaciju vanjskog sustava s modelom digitalnog blizanca.	Python skripta služi kao most između vanjskog, odnosno, stvarnog sustava i digitalnog blizanca, omogućujući razmjenu podataka. Skripta prikuplja podatke (od, npr., senzora), obrađuje ih i prosljeđuje digitalnom blizancu za analizu i donošenje odluka. Ova komponenta osigurava pravovremenu reakciju na promjene u proizvodnom procesu.
5.	Pustiti u pogon proizvodnog digitalnog blizanca.	Implementacija digitalnog blizanca u stvarno proizvodno okruženje uključuje, npr., povezivanje sa senzorima i praćenje stvarnog procesa proizvodnje. Tijekom ove faze provodi se testiranje funkcionalnosti i prilagodba modela prema specifičnostima proizvodnog sustava. Nakon uspješne integracije, proizvodni digitalni blizanac počinje raditi u operativnim uvjetima.
6.	Pratiti izvođenje proizvodnog digitalnog blizanca.	Kontinuirano praćenje izvođenja digitalnog blizanca omogućuje evaluaciju njegove učinkovitosti i preciznosti predikcija. Analiziraju se prikupljeni podaci, uspoređuju se rezultati izvođenja digitalnog blizanca sa simuliranim modelom proizvodnog procesa, odnosno, s očekivanjima, te se identificiraju potencijalna poboljšanja.
6.1	Prikupljati stvarne podatke o izvođenju DB-a iz vanjskog sustava.	Iz vanjskog sustava (npr., iz sustava izvršne proizvodnje) se kontinuirano prikupljaju i pohranjuju podaci o radu strojeva i drugih elemenata proizvodnog procesa. Ovi podaci se koriste za analizu performansi i unapređenje digitalnog blizanca. Točnost i frekvencija prikupljanja podataka važni su kako bi model ostao pouzdan.

6.2.	Usporediti rezultate izvođenja sa simuliranim modelom proizvodnog poslovnog procesa.	Uspoređuju se stvarni rezultati izvođenja digitalnog blizanca s očekivanim izlazima simuliranog modela te se analiziraju odstupanja kako bi se identificirali potencijalni uzroci i prilagodili parametri digitalnog blizanca. Cilj je da model vjerno odražava stvarne uvjete rada.
6.3.	Periodički optimizirati prediktivni model temeljem stvarnih podataka.	Kako bi prediktivni model ostao relevantan, potrebno ga je redovito ažurirati s novim podacima. Proces optimizacije uključuje ponovno treniranje modela, prilagodbu parametara i evaluaciju poboljšanja. Na taj se način postiže dugoročna preciznost predikcija i povećava učinkovitost održavanja.
7.	Optimizirati rad proizvodnog digitalnog blizanca.	Optimizacija rada proizvodnog digitalnog blizanca uključuje periodičku prilagodbu modela i njegovih komponenti kako bi se postigla čim veća efikasnost. Ovisno o analizi izvedbe, mogu se prilagoditi algoritmi, poboljšavati poslovna pravila i/ili integrirati nove tehnologije.

Tablica 4 Aktivnosti metodike razvoja digitalnog blizanca proizvodnog poslovnog procesa



Slika 18 Metodika razvoja proizvodnog digitalnog blizanca

S ciljem detaljnijeg objašnjenja predložene metodike, napravljena je sljedeća tablica 5 koja prikazuje povezani slijed koraka metodike za razvoj proizvodnih digitalnih blizanaca, uz odgovarajuće metode i tehnike te ulaze i izlaze.

Korak u metodici	**Metode i tehnike**	**Ulazi i izlazi**
1. Napraviti analizu proizvodnog poslovnog procesa	Razgovor s radnicima u proizvodnji, analiza dokumenata, analiza tijekova izvođenja u NodeRed-u	Ulazi: postojeći procesi, dokumentacija, sustavi izvršavanja proizvodnje (MES) Izlazi: uvid u ključne aktivnosti, nedostaci, zahtjevi
2. Mapirati aktivnosti proizvodnog poslovnog procesa	BPMN modeliranje, analiza tokova aktivnosti	Ulazi: analiza iz koraka 1, iskustva korisnika Izlazi: BPMN modeli procesa s jasno definiranim ulazima i izlazima
3. Razviti simulirani model proizvodnog poslovnog procesa	IBM WebSphere Business Modeler, simulacijsko modeliranje	Ulazi: BPMN model, vremenski i resursni podaci Izlazi: simulacijski model s mogućnošću testiranja scenarija
4. Razviti proizvodnog digitalnog blizanca	Integracija podataka, povezivanje s realnim sustavima putem IoT uređaja, definiranje komunikacijskih protokola (npr. MQTT), obrada i normalizacija senzorskih podataka, osiguranje semantičke interoperabilnosti, upravljanje pristupom i osnovna sigurnost razmjene podataka.	Ulazi: simulirani model, podaci iz stvarnog sustava Izlazi: funkcionalni digitalni blizanac
4.1. Ugraditi složenije poslovno pravilo	Definiranje pravila odlučivanja	Ulazi: poslovni zahtjevi, scenariji održavanja Izlazi: logika pravila integrirana u model
4.1.1. Razviti prediktivni model za predviđanje održavanja strojeva	Analiza povijesnih podataka, algoritmi strojnog učenja (npr., neuronske mreže)	Ulazi: povijesni podaci, značajke kvarova Izlazi: istrenirani prediktivni model
4.1.2. Trenirati neuronsku mrežu	TensorFlow, Python, duboko učenje	Ulazi: označeni podaci, arhitektura mreže Izlazi: optimizirani model za predikciju kvarova
4.1.3. Ugraditi prediktivni model održavanja u proizvodni digitalni blizanac	Ugradnja prediktivnog modela u model napravljen po BPMN-u u Camundi	Ulazi: istrenirana mreža, procesni model Izlazi: digitalni blizanac s integriranom logikom umjetne inteligencije

4.2. Uključiti komunikacijski element prema vanjskom sustavu	REST API, MQTT, integracija senzora, određivanje protokola komunikacije (npr. HTTPS, WebSocket, MQTT *over* TLS), definiranje semantičke strukture podataka (npr. JSON *schema*), autentifikacija korisnika i uređaja.	Ulazi: zahtjevi za komunikaciju, tehnička infrastruktura Izlazi: povezanost modela sa stvarnim podacima
4.2.1. Izraditi proizvodnog digitalnog blizanca u alatu za procesno servisnu aplikaciju	Camunda BPMN implementacija, korisničke forme	Ulazi: BPMN model, logika poslovnog pravila Izlazi: procesna aplikacija spremna za korištenje
4.2.2. Izraditi Python skriptu za komunikaciju vanjskog sustava s modelom digitalnog blizanca	Python, API pozivi, definiranje JSON/XML strukture razmijenjenih poruka, upravljanje tokovima podataka i validacija ulaza, upravljanje sigurnošću pristupa, konfiguracija protokola za sigurnu komunikaciju (npr. SSL/TLS).	Ulazi: strukturirani podaci iz senzora ili baze Izlazi: podatkovna komunikacija
5. Pustiti u pogon proizvodnog digitalnog blizanca	Implementacija, testiranje, povezivanje sa senzorima	Ulazi: funkcionalni digitalni blizanac, stvarni sustav Izlazi: aktivirani blizanac u stvarnom okruženju
6. Pratiti izvođenje proizvodnog digitalnog blizanca	Praćenje performansi, usporedba sa simulacijom	Ulazi: logovi izvršavanja Izlazi: izvješća o učinkovitosti, odstupanja
6.1. Prikupljati stvarne podatke o izvođenju DB-a iz vanjskog sustava	Automatizirano prikupljanje, spremanje u bazu	Ulazi: izvršni podaci Izlazi: set stvarnih podataka za analizu
6.2. Usporediti rezultate izvođenja sa simuliranim modelom proizvodnog poslovnog procesa	Analiza odstupanja, validacija performansi	Ulazi: stvarni i simulirani rezultati Izlazi: potvrda vjerodostojnosti digitalnog blizanca, preporuke
6.3. Periodički optimizirati prediktivni model temeljem stvarnih podataka	*Retraining* modela, fino podešavanje hiperparametara	Ulazi: novi podaci iz stvarnog sustava Izlazi: ažuriran i precizniji prediktivni model
7. Optimizirati rad proizvodnog digitalnog blizanca	Prilagodba poslovnih pravila, dodavanje funkcionalnosti	Ulazi: rezultati praćenja, analize učinkovitosti Izlazi: poboljšani digitalni blizanac, preporuke za daljnji razvoj

Tablica 5 Slijed koraka metodike za razvoj proizvodnih digitalnih blizanaca, odgovarajućih metoda, ulaza i izlaza

Tablica 5 prikazuje strukturirani prikaz metodike razvoja proizvodnog digitalnog blizanca kroz jasno definirane korake, metode te ulaze i izlaze pojedinih faza. Svaki korak u metodici vezan je uz specifične metode i tehnike koje su korištene u praksi, čime se osigurava transparentnost provedbe i ponovljivost istraživačkog postupka. Tablica omogućuje uvid u to kako se iz početne analize stvarnog

proizvodnog procesa, kroz modeliranje i simulaciju, dolazi do izgradnje funkcionalnog digitalnog blizanca, uključujući integraciju podataka iz stvarnog okruženja i ugradnju prediktivnog modela temeljenog na neuronskim mrežama. Tablica je bitna jer navodi ulaze i izlaze za svaki korak, čime se osigurava logički kontinuitet među fazama te se dodatno potvrđuje iterativni i modularni karakter metodike. Prikaz uključuje i tehničke elemente poput komunikacije s vanjskim sustavima, povezivanja sa senzorima, evaluacije rezultata izvođenja te optimizacije modela. Time se metodika ne zaustavlja na inicijalnoj izradi blizanca, već uključuje njegovu prilagodbu i usavršavanje kroz stvarno korištenje i povratnu analizu.

Razvoj digitalnog blizanca proizvodnog poslovnog procesa u ovom istraživanju temelji se na jasno strukturiranoj metodici koja kombinira poslovno modeliranje, simulacijski eksperiment, primjenu algoritma umjetne inteligencije te integraciju stvarnih podataka u realnom vremenu. Metodika je osmišljena tako da ne slijedi samo linearan niz aktivnosti, već reflektira iterativni karakter razvoja digitalnih rješenja u dinamičnom proizvodnom okruženju. Izvođenje proizvodnog digitalnog blizanca se prati te se po potrebi optimizira, a kako bi se uspješno optimizirao, to uključuje ponovnu analizu proizvodnog poslovnog procesa. Na taj način je omogućeno kontinuirano usavršavanje modela.

U početnim fazama (koraci metodike 1–3), naglasak je na razumijevanju i formalizaciji poslovnog procesa. Prikupljanjem informacija iz proizvodnje, dokumentacije i izravnim razgovorom s radnicima, dobiva se sveobuhvatna slika stvarnog poslovnog procesa. Na temelju toga izrađuje se BPMN model koji vizualizira procesne tokove i omogućava jasno definiranje ulazno-izlaznih parametara. Ova faza daje podlogu i za razvoj simulacijskog modela u alatu za procesne simulacije, čime se omogućava testiranje ponašanja procesa u kontroliranom virtualnom okruženju.

U srednjim fazama (koraci 4–4.2.2), metodika prelazi u tehnički zahtjevniji dio. Izrađuje se funkcionalni digitalni blizanac koji ne samo da replicira procesno ponašanje već uključuje i prijedlog odluka temeljem poslovnih pravila. Ovdje se posebno ističe ugradnja prediktivnog modela temeljenog na neuronskim mrežama, treniranog nad povijesnim podacima o kvarovima strojeva. Umjetna inteligencija se ne koristi kao izolirani modul već je funkcionalno integrirana unutar BPMN modela, kao dio poslovne logike. Povezivanje digitalnog blizanca s vanjskim sustavom, realizirano putem REST API komunikacije i Python skripti, omogućuje primanje podataka u stvarnom vremenu i prijedlog odluka na temelju aktualnog stanja u proizvodnji.

U završnim fazama (koraci 5–7), digitalni blizanac se pušta u pogon, gdje se kontinuirano prate njegova izvedba i točnost predikcija. Analizom odstupanja između stvarnih i simuliranih podataka, metodika omogućuje identifikaciju slabih točaka kako bi se izvršila njihova korekcija. Prediktivni model se periodički optimizira temeljem novih podataka kako bi zadržao svoju točnost i pouzdanost, čime se osigurava dugoročna održivost i primjenjivost cijelog sustava.

U dostupnoj znanstvenoj i stručnoj literaturi ne postoji jedinstvena, standardizirana metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa koja bi obuhvatila sve njegove komponente, uključujući integraciju simulacijskog modela, prediktivnu analitiku i komunikacijske mehanizme u stvarnom vremenu. Postoje određeni okviri koji se odnose na razvoj digitalnih blizanaca u širem smislu, primjerice metodološki pristupi izloženi u radovima poput Fuhrländer-Völker i sur. (2025), kao i smjernice Digital Twin Consortium-a (2024) no oni su najčešće općeg karaktera i ne uključuju konkretne korake prilagođene poslovnim procesima u proizvodnim sustavima. Takvi pristupi

nerijetko ostaju na razini konceptualnih modela ili se usmjeravaju isključivo na tehničke ili proizvodne aspekte, bez uključivanja poslovne logike i prediktivnih pravila temeljenih na stvarnim podacima. U usporedbi s metodikom od (Fuhrländer-Völker, D. i ostali, 2025), koja počiva na teorijskim okvirima, metodika predložena u ovom radu je usmjerena isključivo na digitalne blizance proizvodnih poslovnih procesa, s jasno definiranim koracima koji uključuju i mapiranje aktivnosti, izradu simulacija, ugradnju AI pravila i integraciju s vanjskim sustavima. Fuhrländer-Völker metodika se bavi se općim razvojem digitalnih blizanaca u proizvodnji, ne ulazi duboko u poslovne procese. Naglasak je na konceptima poput digitalnog tijeka podataka, razmjeni modela i standardizaciji među alatima. Metodika u ovom radu uključuje neuronske mreže za prediktivno održavanje, Python skripte za komunikaciju, konkretne alate (IBM WebSphere, Camunda), rad sa sintetičkim i stvarnim podacima. Fuhrländer-Völker metodika općenito navodi potrebu za podatkovnim povezivanjem i interoperabilnošću, ne spominje konkretne algoritme, skripte ili AI pristupe. I posljednje, metodika u ovom radu je visoko tehnički i implementacijski orijentirana i jasno navodi ograničenja (npr. dostupnost podataka) dok Fuhrländer-Völker metodika je više strateški okvir, koji se može generalizirati na više vrsta proizvodnih sustava, ali ne nudi operativnu provedbu na toj razini detalja. Predložena metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa razlikuje se upravo po svojoj specifičnosti: ona je usmjerena na mapiranje i modeliranje poslovnih aktivnosti, ugradnju poslovnih pravila temeljenih na umjetnoj inteligenciji, integraciju s vanjskim sustavima te kontinuirano praćenje i optimizaciju rada digitalnog blizanca u stvarnim i simuliranim uvjetima.

Praćenje izvođenja proizvodnog digitalnog blizanca ključno je za osiguravanje njegove točnosti, učinkovitosti i sposobnosti da u realnom vremenu odražava stanje stvarnog sustava. Kontinuiranim prikupljanjem podataka iz vanjskih izvora omogućuje se analiza stvarnog ponašanja sustava i usporedba s očekivanim rezultatima simuliranog modela. Takva kontrola rada digitalnog blizanca omogućuje identifikaciju odstupanja i njihovih uzroka te prilagodbu parametara modela kako bi on ostao vjeran prikaz stvarnosti. Osim toga, redovita optimizacija prediktivnog modela temeljem novih podataka doprinosi dugoročnoj preciznosti predikcija i većoj učinkovitosti sustava. Na temelju prikupljenih uvida provodi se i šira optimizacija digitalnog blizanca kroz prilagodbu komponenti, poslovnih pravila ili čak uvođenje novih tehnologija, čime se osigurava njegova dugoročna relevantnost i operativna vrijednost u proizvodnom okruženju.

Metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa razvijena je specijalizacijom teorijskih istraživanja i sustavnom primjenom paradigme znanstvenog istraživanja dizajna (DSR) kroz tri ključne faze koje su omogućile njezinu konceptualizaciju i validaciju. U prvoj fazi definiranja problema provedena je detaljna analiza postojećih ograničenja u modeliranju proizvodnih procesa i identificirane su specifične potrebe za razvojem jedinstvenog, standardiziranog pristupa razvoja digitalnog blizanca proizvodnog poslovnog procesa koji bi obuhvatio sve njegove važne komponente, poput prediktivne analitike i komunikacijskih mehanizama u stvarnom vremenu. Druga faza razvoja rješenja uključivala je iterativni pristup kreiranju metodike kroz praktičnu primjenu u stvarnom proizvodnom okruženju, pri čemu su se koraci metodike postupno definirali i rafinirani na temelju empirijskih uvida i povratnih informacija iz prakse. Treća faza verifikacije metodike je provedena kroz simulacijski eksperiment koji je omogućio validaciju metodike usporedbom simuliranih rezultata, napravljenih pomoću povijesnih podataka, s analizom izvedbe proizvodnih digitalnih blizanaca. Kroz ovakav DSR proces, metodika je evoluirala od početne konceptualne ideje do

validiranog skupa koraka i aktivnosti koji omogućavaju sustavni razvoj digitalnog blizanca proizvodnog poslovnog procesa.

U okviru ovog istraživanja, predložena metodika razvoja proizvodnog digitalnog blizanca validirana je primjenom na dva slučaja korištenja iz stvarnih proizvodnih okruženja: proizvodnji dronova i mobilnih uređaja. U oba slučaja korištena je ista metodika, čime je omogućena njezina dosljedna primjena kroz sve faze: od analize stvarnog procesa, izgradnje modela i implementacije digitalnog blizanca, do evaluacije funkcionalnosti modela u praksi. Time je potvrđena izvedivost i korisnost metodike u stvarnim uvjetima, čime se metodika može smatrati validiranom u kontekstu ovog istraživanja.

S metodološkog aspekta, dodatna razina validacije može se ostvariti kada metodiku preuzmu i primijene neovisni korisnici, primjerice drugi istraživači, stručnjaci iz industrije ili organizacije koje razvijaju slične digitalne blizance. Takva eksterna primjena omogućuje provjeru opće primjenjivosti, skalabilnosti i fleksibilnosti metodike u odnosu na različite domene, podatkovne izvore i tehničke zahtjeve. Uobičajeno je da se ovakva eksterna validacija metodike odvija u kasnijim fazama njezina znanstvenog i stručnog prihvaćanja, kroz implementaciju u drugim projektima ili industrijskim rješenjima. U tom smislu, validacija provedena u ovom istraživanju predstavlja početni, ali ključan korak u dokazivanju vrijednosti metodike, dok se puna potvrda njezine univerzalnosti očekuje kroz njezinu daljnju diseminaciju i neovisnu primjenu u različitim kontekstima.

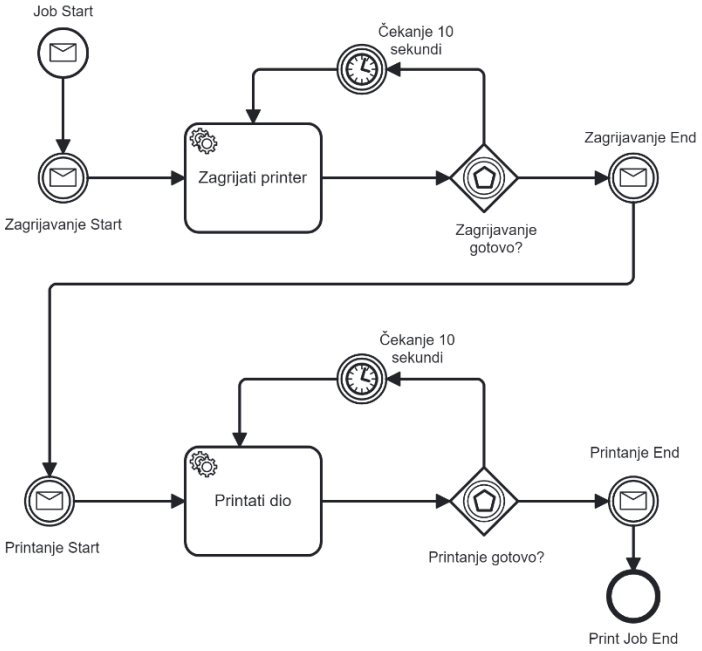
Osobitost predložene metodike je njezina sveobuhvatnost i usmjerenost na integraciju više razina tehnologije: od modeliranja i simulacije, preko algoritama umjetne inteligencije, do komunikacije s fizičkim sustavima. Takav pristup osigurava prilagodljivost, modularnost i ponovljivost u stvarnim uvjetima proizvodnog poslovanja.

14 Digitalni blizanci napravljeni prema novoj metodici razvoja proizvodnih digitalnih blizanaca

14.1 Digitalni blizanac s prediktivnim održavanjem printera

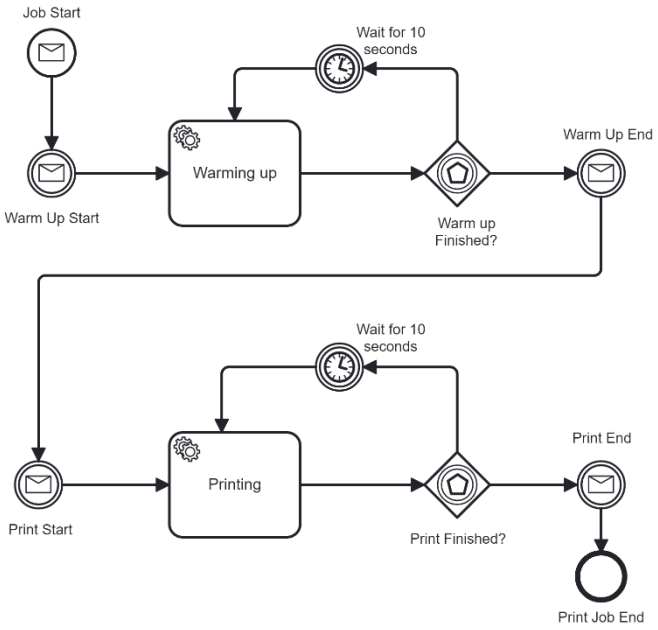
Na slikama 19 i 20 u nastavku se nalazi prikaz digitalnog blizanca poslovnog procesa proizvodnje dronova s informacijom o potrebi održavanja 3D printera, odnosno, radi se o prediktivnom održavanju 3D printera te je ovo aktivnost 4 iz predložene metodike u kojoj se razvija proizvodni digitalni blizanac. Rezultati prediktivnog održavanja temelje se na modelima dviju klasifikacijskih neuronskih mreža razvijenih u programskom jeziku Python te razvoj navedenih neuronskih mreža predstavlja aktivnost 4.1. iz predložene metodike za razvoj proizvodnih digitalnih blizanaca kada se u proizvodni digitalni blizanac ugrađuje složenije poslovno pravilo. Nakon što se zaprimi poruka iz ERP sustava o proizvodnji, radnik odlučuje na kojem printeru će pustiti proizvodnju. Dok radnik pokrene proizvodnju i uđe prva značka (eng., *token*) u proizvodni poslovni proces, pozivaju se neuronske mreže pred trenirane na skupu podataka koje za rezultat šalju da li je potrebno učiniti korektivne radnje nad printerom. Provjerava se da li će printer moći nastaviti proizvodnju sukladno rezultatima strojnog učenja. Ako printeru nisu potrebne korektivne radnje, odnosno, održavanje, proizvodnja se nastavlja te se dio printa. Ako je printeru potrebno održavanje, šalje se informacija kako će printeru trebati održavanje te se proizvodnja nastavlja. U ovom slučaju se radnika upozorava na mogući kvar na printeru prije nego što se on dogodi. Nakon što se dio isprinta, ponovno se ispituje da li je potrebno poduzeti korake održavanja printera. Ako printer neće biti u mogućnosti proizvesti sljedeći dio proizvoda, radnik poduzima korake održavanja printera.

Na slici 19 u nastavku je prikazan model proizvodnog digitalnog blizanaca na hrvatskom jeziku kako bi ga se lakše usporedilo s AS IS modelom napravljenom ranije.



Slika 19 Digitalni blizanac proizvodnje dronova s informacijom o održavanju printera, hrvatska terminologija

Sljedeća slika 20 prikazuje proizvodnog digitalnog blizanca napravljenog u Camundi na engleskoj terminologiji kako bi se lakše pratili rezultati izvođenja proizvodnog digitalnog blizanca koji su također na engleskom jeziku. Ovaj dio predstavlja aktivnost 4.2.1. koja se odnosi na izradu proizvodnog digitalnog blizanca u alatu za procesno servisnu aplikaciju. Radi se o tome što je Camunda napravljena na engleskom jeziku, Python skripte korištene u radu su na engleskom jeziku pa je i model digitalnog blizanca na engleskom jeziku.



Slika 20 Digitalni blizanac proizvodnje dronova s informacijom o održavanju printera, engleska terminologija

U Prilogu 1 ovog dokumenta se nalazi kod neuronskih mreža u Pythonu za održavanje 3D printera i to predstavlja aktivnost 4.1.1. iz predložene metodike kada se razvija prediktivni model za predviđanje održavanja strojeva. Neuronske mreže su izrađene na način da binarni model predviđa postoji li kvar ili ne (*Failure*/*No Failure*), a višeklasni model klasificira vrste kvarova (*Failure*

Types). Treniranje i evaluacija modela se provode na način da se oba modela treniraju na pripremljenim podacima, koristeći unaprijed definirane arhitekture i hiperparametre. Nakon treniranja, modeli se evaluiraju na testnom skupu podataka, a rezultati (točnost) se ispisuju. Python skripta omogućuje učitavanje novih (testnih) podataka iz CSV datoteke, njihovu obradu, te predviđanje vjerojatnosti kvara i vrste kvara. Rezultati predikcija (vjerojatnosti i predviđene vrste kvarova) spremaju se u CSV datoteku. Namjena skripte je da se može koristiti za treniranje modela na povijesnim podacima i za predviđanje kvarova na novim podacima. Dodatno, skripta omogućava automatsko rukovanje kategorijskim i numeričkim obilježjima, skaliranje podataka radi konzistentnosti te detekciju i prijavu eventualnih pogrešaka tijekom obrade testnih podataka.

Kao što je već ranije istaknuto, s obzirom na dostupne podatke o održavanju, napravljene su dvije neuronske mreže koje se međusobno nadopunjavaju. Prva je (1) binarna klasifikacijska mreža koja određuje postoji li rizik od kvara. Ako ne postoji, daljnja analiza nije potrebna. U slučaju da binarna mreža predvidi kvar, nastupa (2) višeklasna klasifikacijska mreža koja identificira vrstu kvara, omogućujući preciznije donošenje operativnih odluka. Ovakav dvofazni pristup pojednostavljuje proces prediktivnog održavanja i osigurava da se računalni resursi fokusiraju na relevantne scenarije koji se mogu dogoditi u poslovnom procesu. Neuronske mreže imaju važnu vremensku komponentu jer mogu zadržati informacije o prethodnim stanjima sustava, što je izuzetno korisno u kontekstu prediktivnog održavanja gdje se traže uzorci u vremenskom slijedu signala koji prethode kvaru. Upravo ta sposobnost prepoznavanja i „pamćenja“ promjena u signalu kroz vrijeme čini neuronske mreže prikladnima za ovakvu primjenu. Detaljniji opis mreža slijedi u nastavku.

(1) Binarna klasifikacijska neuronska mreža s 5 slojeva:

1. Ulazni sloj. Definira ulazni oblik podataka (*input_shape=(input_shape)*), što omogućuje mreži da prihvati značajke poput temperature, istrošenosti alata itd.
2. Gusti sloj sa 64 neurona. U ovom sloju funkcija aktivacije ReLU (Rectified Linear Unit) omogućuje mreži da uči nelinearne relacije u podacima.
3. Isključivi sloj sa stopom od 0.2. Ovaj sloj nasumično isključuje 20% neurona tijekom treniranja, čime se smanjuje prekomjerno prilagođavanje (eng., *overfitting*).
4. Gusti sloj s 32 neurona. Ovo je skriveni sloj s 32 jedinice za daljnje učenje uzorka u podacima.
5. Isključivi sloj sa stopom od 0.2. Koristi se *Dropout* u vrijednosti od 20% kako bi se poboljšala generalizacija modela.
6. Gusti sloj sa 16 neurona. Ovaj sloj dalje sažima informacije kako bi se pripremile za izlaz.
7. Izlazni sloj. *layers.Dense(1, activation='sigmoid')* je izlazni sloj s jednom neuronskom jedinicom i sigmoidnom funkcijom aktivacije. Generira vjerojatnost (0-1) za binarnu klasifikaciju.

Arhitektura neuronske mreže s 5 slojeva omogućila je bolju prilagodbu modela suptilnim obrascima u podacima i stabilniju izvedbu tijekom klasifikacije. U neuronskoj mreži se koriste funkcija *loss='binary_crossentropy'*, koja se koristi za optimizaciju, jer je pogodna za binarne klasifikacijske zadatke te metrika *metrics=['accuracy']*, koja prati točnost tijekom treniranja. Cilj mreže je određivanje hoće li doći do kvara na komponenti stroja ili neće. Ulazi u mrežu su podaci poput temperature zraka, procesne temperature, momenta zakretanja, istrošenosti alata te kategorijske varijable izvedene iz *Product_Type*. Ti podaci se normaliziraju kako bi se osigurala konzistentnost skaliranja i bolja stabilnost tijekom treniranja. Izlaz iz neuronske mreže je jedan neuron sa

sigmoidnom aktivacijskom funkcijom koji predviđa vjerojatnost između 0 i 1. Ako je izlaz blizu vrijednosti 1, predviđa se kvar; ako je blizu vrijednosti 0, predviđa se da neće biti kvara.

Binarna neuronska mreža koristi funkciju gubitka koja koristi *binary_crossentropy* za mjerenje razlike između predviđenih vjerojatnosti i stvarnih binarnih ishoda (0 ili 1). Neuronska mreža je trenirana na označenim podacima (*y_binary*), gdje je svaki primjer označen kao **kvar (1)** ili **bez kvara (0)**. Težine se ažuriraju kroz povratno širenje pogreške (eng., *backpropagation*) kako bi se minimizirao gubitak. Na taj način binarna klasifikacijska neuronska mreža pomaže identificirati da li je potrebno odmah poduzeti održavanje.

(2) Višeklasna (eng., *multiclass*) klasifikacijska neuronska mreža s 5 slojeva:

1. Ulazni sloj, koji kao i kod binarne klasifikacijske neuronske mreže prihvaća ulazne značajke.
2. Gusti sloj s 64 neurona. Početni sloj s 64 neuronske jedinice za učenje kompleksnih uzoraka: *layers.Dense(64, activation='relu')*.
3. Isključivi sloj sa stopom od 0.2. Nasumično isključuje neurone kako bi spriječio *overfitting*: *layers.Dropout(0.2)*.
4. Gusti sloj s 32 neurona. Ovaj sloj koji dalje procesira značajke: *layers.Dense(32, activation='relu')*.
5. Isključivi sloj sa stopom od 0.2. Dodan radi generalizacije.
6. Gusti sloj sa 16 neurona. Završni sloj za daljnju obradu podataka: *layers.Dense(16, activation='relu')*.
7. Izlazni sloj. Izlazni sloj s *num_classes* neuronskih jedinica (po jedan za svaku klasu): *layers.Dense(num_classes, activation='softmax')*. *Softmax* aktivacija generira vjerojatnosti za svaku klasu.

Cilj višeklasne klasifikacijske neuronske mreže je klasificirati vrstu kvara ako se kvar dogodi. Ulazi u neuronsku mrežu su, isto kao kod binarne klasifikacijske neuronske mreže: temperatura zraka, procesna temperatura, moment zakretanja, istrošenost alata te kategorijske varijable izvedene iz *Product_Type*. Izlaz iz neuronske mreže je više neurona u izlaznom sloju (jednako broju vrsta kvarova), pri čemu svaki neuron predstavlja vjerojatnost određene vrste kvara. *Softmax* aktivacijska funkcija osigurava da zbroj svih izlaznih vjerojatnosti bude 1, čime se izražava sigurnost modela za svaku vrstu kvara. Kao i kod binarne klasifikacijske neuronske mreže, postoji funkcija gubitka koja koristi *sparse_categorical_crossentropy* za mjerenje pogreške u predviđanju točne vrste kvara. Neuronska mreža je trenirana na označenim podacima (*y_multiclass*), gdje je svaki primjer označen određenom vrstom kvara (npr. **0** za mehanički kvar, **1** za toplinski kvar). Treniranjem se optimiziraju težine kako bi se maksimizirala točnost predviđanja vrste kvara te ovaj dio predstavlja aktivnost 4.1.2. iz predložene metodike koja se odnosi na treniranje neuronske mreže. Na taj način višeklasna klasifikacijska neuronska mreža pomaže identificirati specifičnu vrstu kvara.

Što se tiče toga kako je skripta implementirana, u skripti se nalazi i funkcija *train_and_evaluate_models()* koja je zadužena za cjelokupan proces treniranja i evaluacije neuronskih mreža. Funkcija prvo učitava i obrađuje podatke koristeći funkciju *load_and_preprocess_data(),* koja provodi kodiranje kategorijskih varijabli, normalizaciju numeričkih značajki i podjelu podataka na trening i testni skup. Nakon toga, kreiraju se i treniraju dva modela: binarni model, koji se trenira kroz 50 epoha s veličinom grupe podataka (eng., *batch size*) od 32, te višeklasni model, koji se također trenira kroz 50 epoha s istim parametrima. Po

završetku treniranja, modeli se evaluiraju na testnom skupu podataka, pri čemu se izračunava i ispisuje točnost svakog modela, omogućujući analizu njihove učinkovitosti u predviđanju kvarova. Još jedna važna funkcija u skripti je *make_prediction()* koja omogućuje predviđanje statusa stroja na temelju novog retka podataka. Ova funkcija vraća JSON objekt koji sadrži vjerojatnost kvara i predviđenu vrstu kvara koji su dio prediktivnog modela proizvodnog digitalnog blizanca te je to aktivnost 4.1.3. iz predložene metodike u kojoj se ugrađuje prediktivni model održavanja u proizvodni digitalni blizanac. Što se funkcionalnosti skripte tiče, zadnja važna stvar za istaknuti je da skripta implementira REST API pomoću *Flask*-a, omogućujući dohvaćanje predikcija putem HTTP zahtjeva. Ključni *endpoint /printer_status* dostupan je putem *GET* metode i vraća JSON objekt koji sadrži vjerojatnost kvara i status stroja, pri čemu status može biti "*OK*" ili "*WARNING*", ovisno o procijenjenom riziku od kvara. Ovime je omogućeno daljinsko korištenje istreniranih modela, čime se predikcije mogu automatski integrirati u digitalne blizance proizvodnih sustava, olakšavajući donošenje odluka u procesu održavanja.

Za potrebe testiranja istreniranih klasifikacijskih neuronskih mreža korištena su dva skupa podataka. Prvi skup podataka sadrži obilježja: *Product ID, Air temperature [K], Process temperature [K], Torque [Nm]* i *Tool wear [min]*, koja predstavljaju ulazne parametre ključne za predikciju kvarova i omogućuju evaluaciju binarne i višeklasne klasifikacije na temelju stvarnih tehničkih parametara procesa. Drugi skup podataka sadrži obilježja: *Product ID, Failure Probability* i *Predicted Failure Type*. Ovaj skup podataka koristi se za usporedbu rezultata modela s očekivanim vrijednostima i za validaciju performansi modela.

Prikazi dijelova skupova podataka za testiranje obje klasifikacijske neuronske mreže se nalaze na slikama 21 i 22 u nastavku. Ovi testni skupovi služe za provjeru točnosti i performansi oba predikcijska modela istovremeno, odnosno, obiju klasifikacijskih neuronskih mreža. Testni skup sadrži 872 slogova.

	A	B	C	D	E	F
1	Product ID	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
2	M12345	298.1	308.6	1551	42.8	0
3	L67890	298.2	308.7	1408	46.3	3
4	H11111	298.3	308.8	1498	49.4	5
5	M54321	298	308.5	1600	43.2	2
6	L98765	298.4	308.9	1450	45.8	4
7	H22222	298.2	308.7	1525	48.1	6
8	L47249	298.9	309	1410	65.7	191
9	M16359	298.1	308.8	2206	16.3	195
10	M16360	298	308.7	2101	17.4	198
11	M16361	298	308.7	1436	48.2	201
12	L48682	298	308.7	1462	48.2	204
13	L48683	298	308.7	1771	26.2	206
14	L48684	298	308.7	1506	47	208
15	L48685	298	308.7	1626	36.7	210
16	L48686	298	308.8	1610	33.7	212
17	M16367	298	308.6	1376	55.7	214
18	M16368	298	308.6	1507	37.3	217
19	L48689	298	308.5	1429	37.7	220
20	M16370	298.1	308.7	1547	38.9	0
21	L48691	298.1	308.7	1642	30.3	3
22	H30926	298	308.7	1503	36.3	5
23	L48693	298.1	308.7	1568	32.9	10
24	M16374	298.1	308.8	1628	37.6	12
25	M16375	298.2	308.9	1614	35.5	15
26	L48696	298.2	308.9	1278	54	18

Slika 21 Isječak iz dijela skupa podataka za testiranje predikcija o održavanju s obilježjima

	A	B	C
1	Product ID	Failure Probability	Predicted Failure Type
2	M12345	0.000216079	No Failure
3	L67890	0.000212521	No Failure
4	H11111	0.001784777	No Failure
5	M54321	0.000489776	No Failure
6	L98765	0.000267056	No Failure
7	H22222	0.001707158	No Failure
8	L47249	0.99839574	Power Failure

Slika 22 Prikaz dijela skupa podataka za testiranje predikcija o održavanju s vjerojatnosti kvara

Smatra se da broj parametara u neuronskoj mreži ovisi o strukturi slojeva i broju veza među neuronima, pri čemu se u obzir uzimaju težine i parametri pristranosti. U praksi se često koristi heurističko pravilo prema kojem skup podataka za treniranje neuronske mreže treba biti otprilike 10 puta veći od ukupnog broja parametara mreže kako bi se izbjeglo preprilagođavanje.

U ovom primjeru, svaka neuronska mreža ima tri gusto povezana sloja s 64, 32 i 16 neurona. Ukupni broj parametara za binarni klasifikator može se izračunati kao:

- Prvi sloj: $(64\times(n+1))$, gdje je n broj ulaznih značajki.
- Drugi sloj: $(32\times(64+1))$.
- Treći sloj: $(16\times(32+1))$.
- Izlazni sloj: $(1\times(16+1))$.

Uz pretpostavku da mreža ima n=5 ulaznih značajki, binarni model ima 4.833 parametara, dok skup podataka ima 10.000 slogova, što zadovoljava pravilo omjera veličine skupa podataka i broja parametara. Razlog za korištenje binarne i višeklasne klasifikacijske neuronske mreže je zbog prirode problema te, sukladno tome, strukture podataka korištenih u istraživanju, koji su odabrani da odgovaraju problemu. Klasifikacija se koristi za probleme u kojima se uzorci dodjeljuju jednoj ili većem broju klasa. Klasifikacijske neuronske mreže se koriste za nadgledano učenje pri čemu su dostupni poznati primjeri s pripadajućim klasama koje model treba naučiti prepoznavati (Laurene Fausett, 1994). Višeslojne neuronske mreže postale su popularan alat za sve širi spektar primjena. Koriste se u robotici, u zadacima prepoznavanja govora ili uzoraka, kodiranju i slično (Raul Rojas, 1996). Binarna klasifikacijska neuronska mreža u ovom slučaju korištenja predviđa da li će se ispad dogoditi ili neće. Binarna klasifikacija je dobar izbor zato što ciljana varijabla *y_binary* eksplicitno definira binarni ishod (npr. 0 za "nema kvara" i 1 za "kvar"). Ovo je prikladno za određivanje prisutnosti ili odsutnosti kvara, što je ključni dio prediktivnog održavanja. Razlog zašto tu ne bi odgovarala neka druga neuronska mreža je taj što primjer uključuje diskretne kategorije, a ne kontinuirane vrijednosti pa, npr., regresijska neuronska mreža nije prikladna. Višeklasna klasifikacijska neuronska mreža u slučaju korištenja predviđa tip kvara (između nekoliko kategorija). Razlog zašto je višeklasna klasifikacijska neuronska mreža dobar izbor je taj što ciljana varijabla *y_multiclass* bilježi različite tipove kvarova (npr., kvar uslijed napajanja), a predviđanje tipova kvarova pomaže u preciznom određivanju specifičnih radnji održavanja. Ovom primjeru tako ne bi odgovarala regresijska neuronska mreža zato što su tipovi kvarova kategorizirani, a ne kontinuirani. Zaključno, regresijsku neuronsku mrežu ne koristimo u ovom primjeru jer regresijska neuronska mreža predviđa kontinuirane numeričke vrijednosti, poput temperature ili stope trošenja, što nije u skladu s primjerom u kojem se predviđaju kvarovi. Nadalje, rekurentne neuronske mreže (RNN) su

prikladne za vremenske nizove. Ako bi skup podataka uključivao sekvence (npr. vremensko ponašanje stroja), RNN bi se mogle koristiti. Međutim, skup podataka koji je uzet za primjer ne obrađuje sekvence već je usmjeren na uvjete rada strojeva što primjenjujemo na 3D printere, bušilicu i prešu što čini RNN neodgovarajućima. Rekurentna mreža je međusobno povezana kompetitivna mreža u kojoj postoje signalni putevi u zatvorenoj petlji koji se vraćaju na istu jedinicu. Znači, u rekurentnoj mreži jedinice su međusobno povezane na način da tvore zatvorene petlje (Laurene Fausett, 1994). Rekurentne mreže omogućuju rješavanje problema koji zahtijevaju obradu ulaza promjenjive duljine, primjerice pri zbrajanju dvaju binarnih brojeva koji se unose u mrežu bit po bit, dok mreža zauzvrat postupno generira bitove rezultata. Takve su mreže dizajnirane tako da ponovno koriste djelomične izračune unutar vlastite strukture. Petlje u njihovoj topologiji omogućuju privremenu pohranu i naknadnu uporabu signala nakon što su generirani (Raul Rojas, 1996).

U Prilogu 2 ovog dokumenta se nalazi Python kod za pripremu baze podataka u Accessu. Da bi proizvodni digitalni blizanac funkcionirao, potreban nam je izvor podataka o trenutnom stanju stvarnog procesa. To se može napraviti na nekoliko načina:

- da se čita stanje sa senzora direktno iz procesa
- na indirektan način poput snimanja procesa kamerom pa se analiziraju slike procesa ili
- spajanjem na registar podataka u poslovnom procesu poput baze podataka iz koje se čitaju podaci. Baza podataka reflektira stanje procesa i to se inače naziva čitanje baze (eng., *polling*).

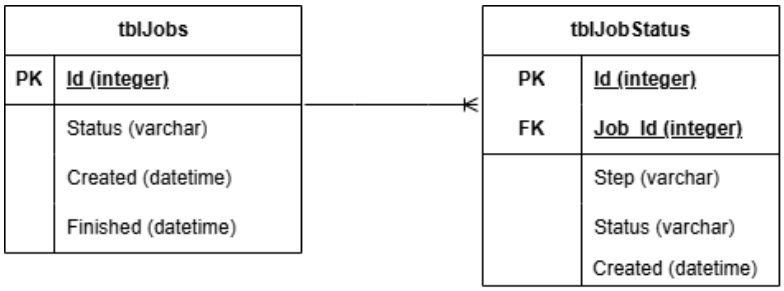
Znači, u ovom istraživanju se radi *polling* baze podataka koja reflektira stanje procesa jer se u bazi evidentiraju pojedine faze procesa koje se odvijaju i vrijeme koliko one traju.

Python skripta priprema Microsoft Access bazu podataka koja se koristi za upravljanje poslovnim procesom proizvodnje dronova. Skripta obavlja četiri zadatka: postavljanje radnog direktorija, povezivanje s bazom podataka, stvaranje potrebnih tablica te osiguravanje ispravne strukture podataka. Prvi korak skripte je postavljanje radnog direktorija (eng., *set_working_directory()*), koji osigurava da se svi podaci i baza podataka nalaze u unaprijed definiranoj mapi. Nakon toga, skripta uspostavlja vezu s Microsoft Access bazom podataka (eng., *connect_to_database()*). Koristi se *pyodbc driver* kako bi se ostvarila veza s datotekom baze podataka *biel.accdb*, čime se omogućuje daljnja manipulacija podacima kroz SQL naredbe.

Glavni zadatak skripte je upravljanje tablicama baze podataka. Definirane su dvije ključne tablice:

- *tblJobs* – Tablica koja prati narudžbe, odnosno, zadatke (eng., *Jobs*) u proizvodnom procesu, pri čemu svaki zadatak ima jedinstveni ID, status, datum i vrijeme kreiranja te eventualni završetak.
- *tblJobStatus* – Tablica koja bilježi statusne promjene svakog zadatka, uključujući korake u procesu (kolona *Step* u tablici), njihov trenutni status (kolona *Status* u tablici) i vrijeme kada su nastali (kolona *Created* u tablici).

Na slici 23 se nalazi ERA dijagram radi prikaza strukture podataka iz realnog sustava i odnosa među entitetima.



Slika 23 ERA dijagram tblJobs i tblJobStatus

Skripta najprije briše postojeće tablice ako one već postoje (naredba *drop_table()*), a zatim ih ponovno kreira (naredba *create_table()*). Ova metoda omogućava da se uvijek koristi čista i ažurirana baza podataka, što je važno za testiranje unutar proizvodnog digitalnog blizanca.

Pošto stvarni proces evidentira u bazi podataka stanje printera (njegovu zauzetost), za potrebe ovog istraživanja, implementiran je simulator proizvodnog procesa u Python skripti koji služi za simulaciju rada 3D printera i bilježenje njegovog statusa u bazi podataka, pri čemu je omogućeno testiranje i validacija digitalnog blizanca bez potrebe za fizičkim sustavom. Budući da stvarni proizvodni proces kontinuirano bilježi stanje printera i njegovu zauzetost u bazi podataka, simulator replicira ovaj mehanizam tako što generira sintetičke podatke i upisuje ih u bazu na isti način kao i stvarni sustav. Znači, simulator simulira stvarni poslovni proces u kontekstu baze podataka da si olakšamo testiranje u trenucima kada nam stvarni fizički sustav nije dostupan u Inovacijskom parku. Implementacija je osmišljena na način da je u Inovacijskom parku snimljen proces i isproban digitalni blizanac na stvarnim podacima u realnom vremenu te je nakon toga proces simuliran lokalno kako bi se prikazalo da proizvodni digitalni blizanac radi, odnosno, replicira ponašanje stvarnog fizičkog sustava. Ovaj princip omogućava testiranje i optimizaciju digitalnog blizanca u kontroliranom okruženju. Simulacija proizvodnog procesa osmišljena je kao dio eksperimentalnog okvira za evaluaciju digitalnog blizanca, pri čemu se njegova točnost i funkcionalnost provjeravaju kroz usporedbu simuliranih podataka s podacima iz stvarnog sustava. Važno je još jednom istaknuti kako je inicijalno stvarni proces snimljen u stvarnom okruženju kako bi se osigurala njegova ispravnost. Nakon toga, isti proces je repliciran u eksperimentalnom okruženju kako bi se potvrdila valjanost modela digitalnog blizanca. Što se tiče dijela prediktivnog održavanja, kao što je već rečeno ranije, podaci korišteni za treniranje prediktivnog modela dolaze iz otvorenog izvora iz razloga što je često nemoguće prikupiti dovoljno veliku količinu podataka nad kojom bi se mogla trenirati umjetna inteligencija.

U Prilogu 3 ovog dokumenta je priložena Python skripta koja implementira objašnjeni simulator proizvodnog procesa. Skripta modelira poslovni proces koji se sastoji od nekoliko aktivnosti, upravlja printerom i bilježi status svake aktivnosti u Microsoft Access bazu podataka. Pri inicijalizaciji, skripta definira put do baze podataka, listu aktivnosti proizvodnog procesa i intervale korištenja resursa. Simulirani proces sastoji se od dvije ključne aktivnosti: zagrijavanja printera (eng., „*Warm Up*“) i printanja (eng., „*Print*“), pri čemu svaka od njih ima specifično trajanje. Skripta također koristi semafore kako bi osigurala da samo jedan zadatak može koristiti određeni resurs u danom trenutku, čime se simulira ograničena dostupnost proizvodnih resursa.

Sustav implementiran u skripti koristi *FastAPI* koji omogućava pokretanje novih poslova (eng., *Jobs*) kroz REST API poziv. Kada korisnik putem API-ja pošalje zahtjev za pokretanjem posla (/start_job),

generira se novi *job ID*, a zatim se pokreće simulacija izvršavanja tog posla. Tijekom izvođenja, status svakog posla bilježi se u bazu podataka pomoću posebnih funkcija koje upravljaju zapisima u tablicama *tblJobs* i *tblJobStatus*. Svaki posao prolazi kroz definirane faze, odnosno, aktivnosti, pri čemu se na početku i završetku svake faze bilježi status u bazu podataka. Jedna od ključnih funkcionalnosti skripte je obavještajni mehanizam, koji omogućuje slanje obavijesti Camundi putem asinkronog reda (eng., *notification_queue*). Ovaj mehanizam osigurava da Camunda može pratiti napredak poslova i na temelju toga inicirati daljnje korake u proizvodnom sustavu. Skripta također uključuje funkcionalnost inicijalizacije brojača, *JOB_ID_COUNTER* prilikom pokretanja (*startup_event()*), čime se osigurava kontinuitet u dodjeljivanju ID-ova poslovima, čak i nakon ponovnog pokretanja procesa proizvodnje. Na kraju, kada se pokrene, *FastAPI* aplikacija koristi *Uvicorn* poslužitelj za dostupnost API-ja na mreži, čime se omogućava interakcija s drugim sustavima u stvarnom vremenu.

Kako bi se periodički mogla čitati baza podataka u kratkim vremenskim intervalima s ciljem otkrivanja promjena u procesu, napravljena je Python skripta pod nazivom *biel_listener* čiji je kod dostupan u Prilogu 4 ovog dokumenta. Ovaj postupak se engleski naziva „*polling*“. Što su intervali „osvježavanja“ baze kraći, to je digitalni blizanac precizniji u praćenju stvarnog stanja. Osluškivač (eng., *listener*) je, znači, komponenta ovog istraživanja koja redovito dohvaća relevantne podatke iz baze i prosljeđuje ih proizvodnom digitalnom blizancu unutar Camunde. Na taj način Camunda odražava stvarno stanje procesa dok se on izvodi. Tako je navedena Python skripta zapravo ključni dio proizvodnog digitalnog blizanca unutar Camunde te je to aktivnost 4.2.2. iz predložene metodike kada se izrađuje Python skripta za komunikaciju stvarnog sustava s modelom digitalnog blizanca. Važno je napomenuti kako simulator implementiran ranije nema direktne veze s izvođenjem digitalnog blizanca. Proizvodni digitalni blizanac u suštini čine osluškivač i BPMN model u Camundi. Osluškivač prati promjene u bazi i obavještava Camundu o tim promjenama, dok proizvodni digitalni blizanac vizualno prikazuje stanje procesa putem Camunde.

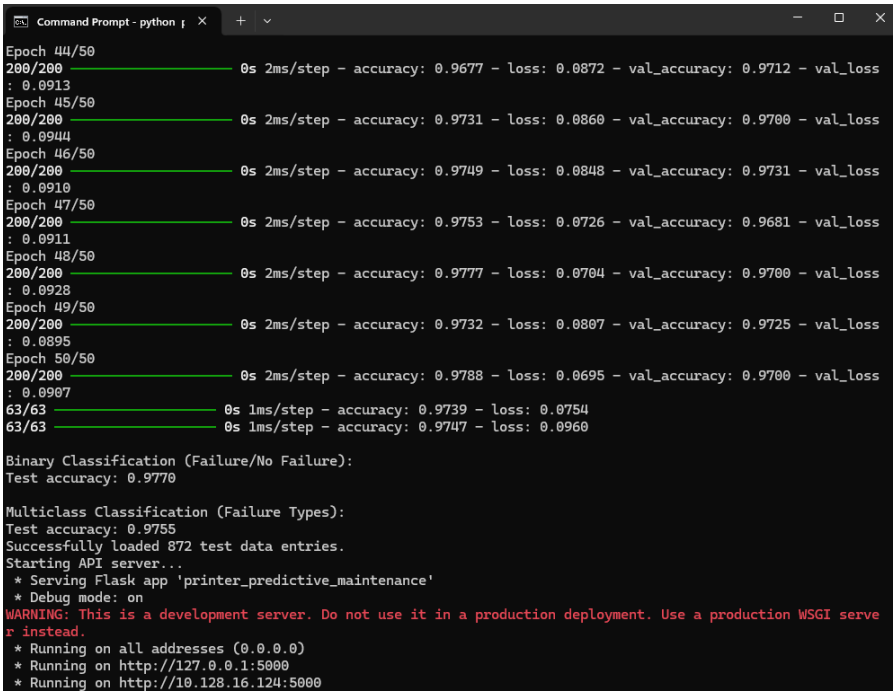
Skripta *biel_listener.py* iz Priloga 4 ovog dokumenta implementira sustav za nadgledanje statusa proizvodnih aktivnosti i njihovu integraciju s Camundom. Glavna funkcija skripte je praćenje tablice *tblJobStatus* unutar baze podataka, prepoznavanje novih zapisa koji označavaju promjene u statusima proizvodnih poslova te slanje odgovarajućih obavijesti Camundi putem REST API-ja. Na ovaj način osigurava se sinkronizacija između baze podataka koja bilježi tijek proizvodnje i poslovnog procesa koji se izvršava unutar Camunde.

Skripta pokreće *FastAPI* Web aplikaciju. Pri pokretanju aplikacije inicijaliziraju se dva ključna pozadinska zadatka: radnik za slanje obavijesti Camundi (eng., *notify_camunda_worker*) i monitor statusa (eng., *status_monitor*). Monitor statusa periodički pretražuje tablicu *tblJobStatus* kako bi identificirao nove zapise o promjenama u statusima poslova. Ako se otkrije novi unos koji prethodno nije obrađen, kreira se odgovarajuća poruka koja se stavlja u asinkroni red obavijesti (eng., *notification_queue*). Zatim radnik preuzima poruku iz reda i šalje je Camundi putem HTTP zahtjeva.

Integracija s Camundom odvija se putem REST API-ja, gdje se za svaki promijenjeni status posla kreira poruka koja može inicirati ili ažurirati poslovni proces u Camundi. Ako je riječ o početku posla, skripta šalje poruku s procesnim varijablama, dok se za sve ostale korake koristi korelacijski ključ (eng., *correlationKeys*), čime se osigurava povezivanje promjena statusa s odgovarajućim poslovnim procesom. Ovaj pristup se koristi zato da se omogući automatizirano pokretanje daljnjih radnji unutar

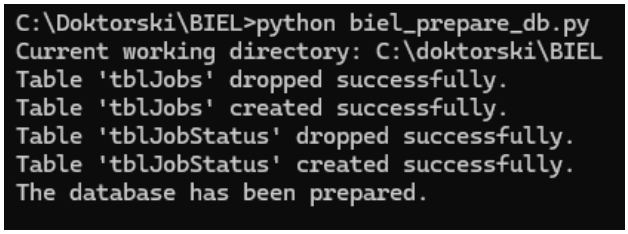
poslovnog procesa u Camundi, čime se postiže potpuna digitalna integracija između proizvodnog sustava i upravljačkog procesa. Važnost ove skripte je u tome što omogućuje realnu sinkronizaciju podataka o proizvodnom procesu s procesnim upravljanjem u okviru digitalnog blizanca. Time se osigurava da se svaka promjena u proizvodnom procesu automatski reflektira, što je značajno za optimizaciju proizvodnog tijeka, pravovremenu reakciju na događaje u proizvodnji i prediktivno održavanje.

Slika 24 u nastavku prikazuje stanje u terminalu uspješno trenirane neuronske mreže. Na slici je vidljivo da se model trenira kroz 50 epoha, pri čemu svaka epoha sadrži 200 iteracija. Tijekom treniranja prikazuju se ključne metrike poput točnost (eng., *accuracy*) i gubitka (eng., *loss*) na treniranju i skupu za testiranje (validaciju). Prema prikazu, završna trenirana točnost doseže 97.8%, dok je validacijska točnost oko 97.0%. Nakon treniranja, model se testira na zasebnom skupu podataka. Rezultati pokazuju da neuronska mreža postiže 97.7% točnosti u binarnoj klasifikaciji (*Failure*/*No Failure* objašnjeni ranije) i 97.55% točnosti u višeklasnoj klasifikaciji (*Failure Types* objašnjeni ranije). Učitano je 872 testna primjera koji se koriste za evaluaciju modela. Skripta pokreće Flask API server za prikazani model pod nazivom "*printer_predictive_maintenance*".



Slika 24 Prikaz uspješno treniranih neuronskih mreža iz terminala s 872 generiranih testnih zapisa

Slika 25 u nastavku prikazuje uspješno kreiranu bazu podataka pomoću Python skripte „*biel_prepare_db*“, koja sadrži tablice *tblJobs* i *tblJobStatus*.



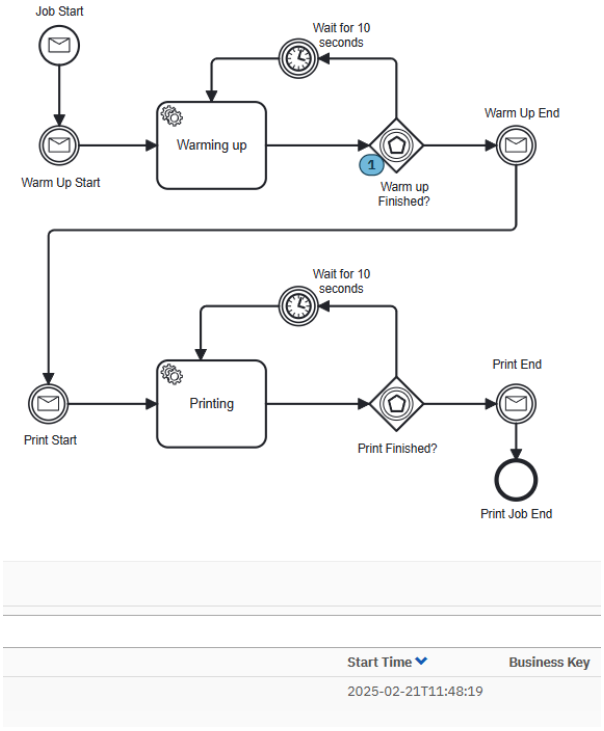
Slika 25 Prikaz uspješno kreirane baze podataka pomoću Python skripte „biel_prepare_db“

Da je posao (*Job*) iniciran, prikazano je na slici 26 u nastavku koja reprezentira da je pokrenuto zagrijavanje printera koje traje 20 minuta i ovo je aktivnost 5 iz predložene metodike kada se proizvodni digitalni blizanac pušta u pogon.

```
  @app.on_event("startup")
INFO:     Started server process [17780]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8001 (Press CTRL+C to quit)
[2025-02-21 11:48:18.645014] Job 1 : Job Start
[2025-02-21 11:48:21.971119] Job 1 : Warm Up Start
```

Slika 26 Iniciran posao (eng., Job), prikaz statusa resursa u terminalu listenera

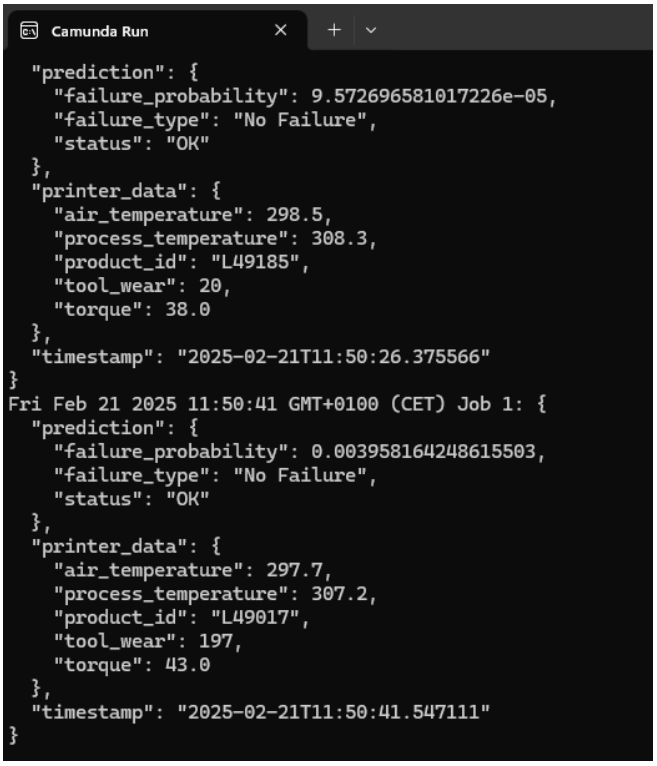
Camundina platforma također prikazuje instancu procesa, odnosno, koji posao se trenutno izvodi. Na slici 27 je vidljivo kako traje 20-minutno zagrijavanje printera te se čeka i redovito (svakih 10 sekundi) ispituje da li je printer zagrijan.



Slika 27 Prikaz pokrenutog zagrijavanja printera (Job 1) u Camundinoj platformi

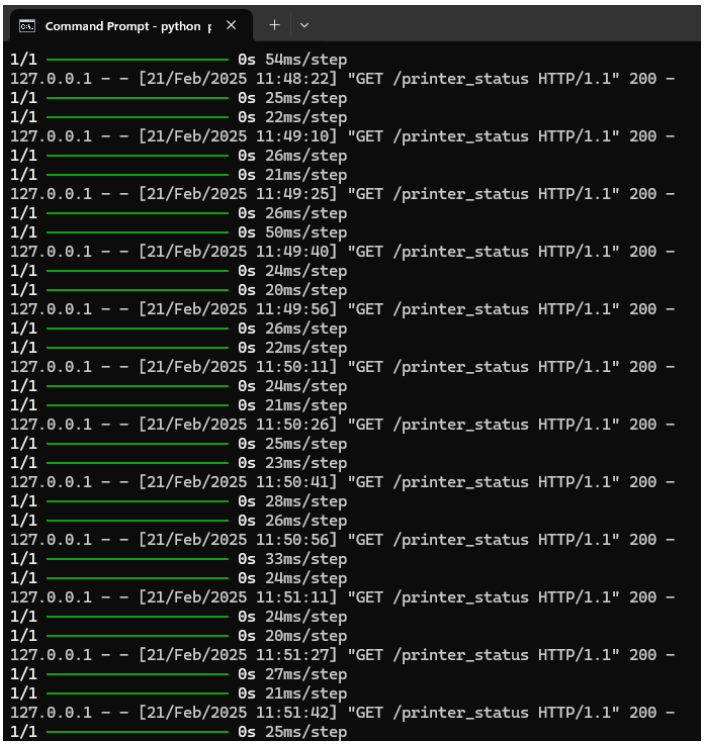
Osim u Camundinoj platformi, izvođenje procesa se može pratiti i u konzoli od Camunde kao što je to prikazano na slici 28 u nastavku. Na slici 28 je prikazan izlaz iz Camundine konzole koji prikazuje podatke o predikciji kvara printera i parametrima uređaja. Svaka zabilježena iteracija sadrži predikciju kvara, gdje se prikazuje vjerojatnost kvara (eng., *failure_probability*), klasifikacija tipa kvara (eng., *failure_type*) i trenutni status uređaja (npr. "OK"). Uz predikcijske podatke, prikazuju se i podaci printera, koji uključuju temperaturu zraka, temperaturu procesa, istrošenost alata i moment zakretaja. Također, zabilježene su vremenske oznake (eng., *timestamp*) koje označavaju kada su podaci generirani. Navedeno je aktivnost 6.1. iz predložene metodike koja se odnosi na prikupljanje stvarnih podataka o izvođenju digitalnog blizanca iz vanjskog sustava. Uz svaki zapis vidi se da

Camunda obrađuje posao, što sugerira da proces praćenja rada printera kontinuirano radi i ažurira stanje sustava u stvarnom vremenu.



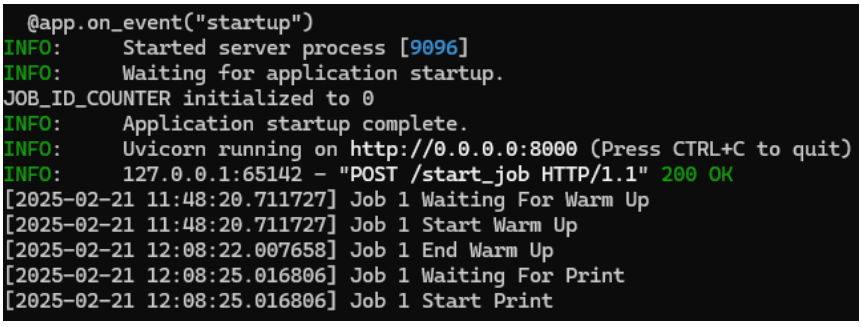
Slika 28 Prikaz kako konzola Camunde također prima podatak o pokrenutom procesu, odnosno, zagrijavanju printera

Slika 29 u nastavku prikazuje izvođenje Python skripte za treniranje neuronskih mreža, gdje se periodički šalju *HTTP GET* zahtjevi prema API *endpointu /printer_status*. Vremenske oznake pokazuju da se zahtjevi izvode u pravilnim intervalima, sugerirajući da se radi o *polling* mehanizmu, odnosno kontinuiranom dohvaćanju statusa printera. Također, u terminalu su prikazane informacije o vremenu izvršavanja svakog koraka (u milisekundama po koraku).

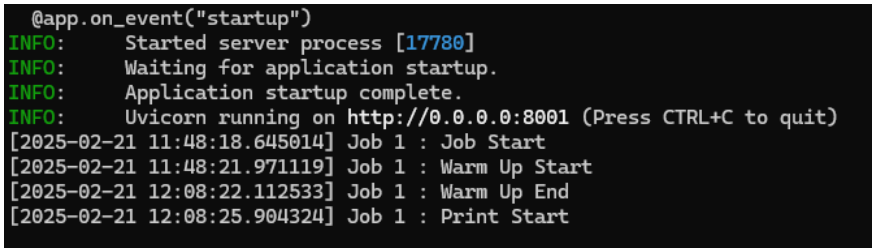


Slika 29 Izvođenje neuronske mreže dok traje zagrijavanje printera

Početak proizvodnje, odnosno, printanja dijela drona se vidi na slikama 30 i 31 u nastavku koje prikazuju status posla, a time i status proizvodnje. Nakon zagrijavanja printera, započinje proizvodnja koja traje otprilike sat i 30 minuta.

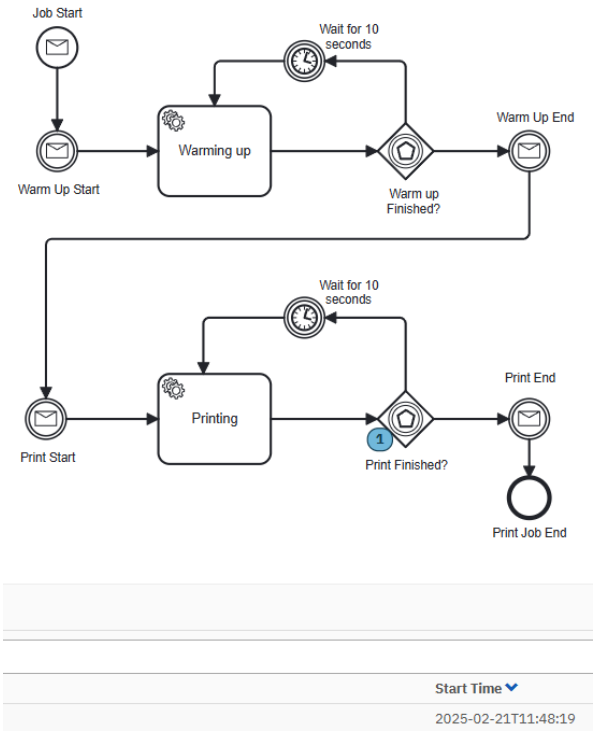


Slika 30 Status procesa, trenutak pokrenute proizvodnje u terminalu simulatora



Slika 31 Status korištenja resursa u trenutku pokrenute proizvodnje u terminalu listenera

Proizvodni digitalni blizanac i njegovo izvođenje se vidi u platformi Camunde na slici 32 u nastavku. Slika u nastavku prikazuje kako se odvija printanje te se periodički šalje upit da li je proces proizvodnje gotov i primaju se podaci prediktivnog modela. Znači, čekanje od 10 sekundi u modelu je i radi dobivanja podataka prediktivnog modela o statusu printera.

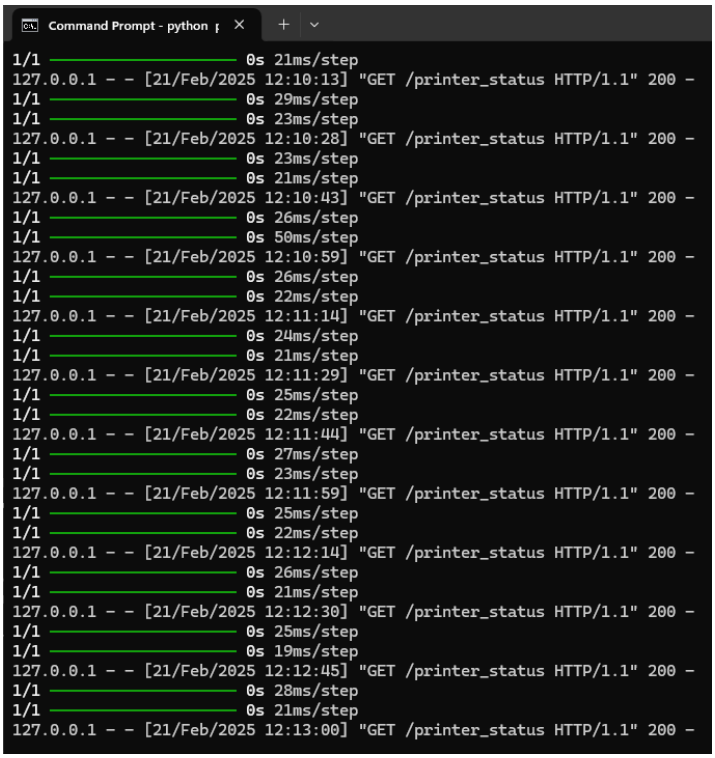


Slika 32 Prikaz početka proizvodnje dijela drona u Camundi

Priložena slika 33 u nastavku prikazuje konzolu Camunde s prikazom periodičkih podataka o predikciji mogućih kvarova i parametrima printera tijekom izvršavanja posla 1. Svaka iteracija sadrži predikciju kvara, što uključuje vjerojatnost kvara, tip kvara i trenutni status uređaja (u svim slučajevima "OK"). Također, zapisani su parametri printera, poput temperature zraka i procesa. Svaka serija podataka sadrži i vremensku oznaku, koja označava trenutak prikupljanja podataka. Ova slika govori da Camunda kontinuirano prati stanje printera tijekom printanja dijela, dok neuronske mreže analiziraju podatke i procjenjuju mogućnost kvara (aktivnost 6.1. iz predložene metodike).

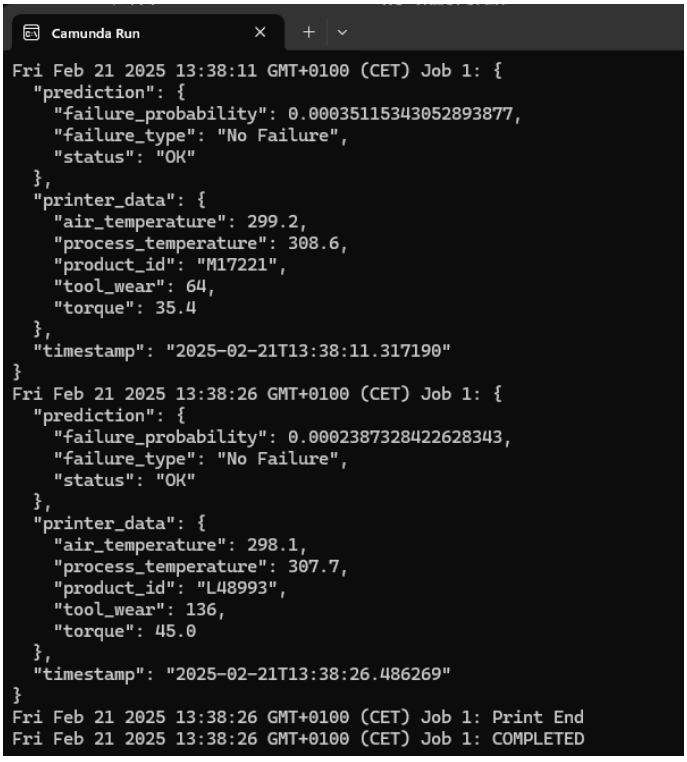
Slika 33 Stanje u konzoli Camunde nakon što je započela proizvodnja dijela drona

U produžetku se nalazi prikaz treniranja neuronskih mreža dok traje proizvodnja, kako bi se naglasila važnost pravovremenih podataka tijekom cijelog procesa proizvodnje.



Slika 34 Treniranje neuronske mreže dok traje proizvodnja

Uspješan završetak procesa proizvodnje je prikazan na slici 35 u nastavku koja prikazuje rezultat u konzoli Camunde te se na njoj vidi vrijeme završetka printanja, odnosno, posla. Sa završetkom printanja dijela drona, završava i proizvodnja pošto je pokrenuta jedna instanca procesa iz razloga što proizvodnja duže traje.



Slika 35 Prikaz završetka procesa proizvodnje u konzoli od Camunde

Slika 36 u nastavku prikazuje redoslijed korištenja resursa tijekom proizvodnje te posao po svakoj fazi korištenja resursa. Ovo je prikaz nakon što je proizvodnja završila kako bi se vidjelo trajanje zauzetosti resursa po poslu.

```
  @app.on_event("startup")
INFO:     Started server process [17780]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8001 (Press CTRL+C to quit)
[2025-02-21 11:48:18.645014] Job 1 : Job Start
[2025-02-21 11:48:21.971119] Job 1 : Warm Up Start
[2025-02-21 12:08:22.112533] Job 1 : Warm Up End
[2025-02-21 12:08:25.904324] Job 1 : Print Start
[2025-02-21 13:38:26.388089] Job 1 : Print End
[2025-02-21 13:38:31.124651] Job 1 : Job End
```

Slika 36 Status posla i redoslijed korištenja resursa tijekom proizvodnje nakon što je ista završila

Osim zauzetosti resursa, u konzoli je moguće pratiti redoslijed izvođenja aktivnosti tijekom proizvodnje po poslu. To je prikazano na slici 37 u nastavku, koja prikazuje točna vremena od trenutka kada je počelo zagrijavanje printera, koje sve aktivnosti su bile uključene tijekom izvođenja procesa do trenutka kada je proces proizvodnje završio. Ukupan proces proizvodnje je trajao 1 sat i 50 minuta. U to vrijeme spada vrijeme zagrijavanja printera i vrijeme printanja. Kao što je već ranije rečeno, pošto proces proizvodnje duže traje, za ovu eksperimentalnu priliku je puštena samo jedna proizvodnja. Kada bi rad proizvodnog digitalnog blizanca bilo potrebno optimizirati, to bi se učinilo u ovom trenutku, prema predloženoj metodici i aktivnosti 7 koja govori o optimizaciji rada digitalnog blizanca.

```
  @app.on_event("startup")
INFO:     Started server process [9096]
INFO:     Waiting for application startup.
JOB_ID_COUNTER initialized to 0
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     127.0.0.1:65142 - "POST /start_job HTTP/1.1" 200 OK
[2025-02-21 11:48:20.711727] Job 1 Waiting For Warm Up
[2025-02-21 11:48:20.711727] Job 1 Start Warm Up
[2025-02-21 12:08:22.007658] Job 1 End Warm Up
[2025-02-21 12:08:25.016806] Job 1 Waiting For Print
[2025-02-21 12:08:25.016806] Job 1 Start Print
[2025-02-21 13:38:26.869273] Job 1 End Print
[2025-02-21 13:38:29.884590] Job 1 COMPLETED
```

Slika 37 Redoslijed izvođenja aktivnosti tijekom procesa proizvodnje u terminalu

U bazi je također moguće pratiti broj puštenih instanci i to u tablici *tblJobs* gdje je vidljiv broj puštenih poslova, status tih poslova, vrijeme kada je posao počeo i vrijeme kada je posao završio.

tblJobs

Id	Status	Created	Finished	Click to Add
1	COMPLETED	21/02/2025 11:48:16	21/02/2025 13:38:30	

Slika 38 Stanje u bazi, kada je proces proizvodnje započeo, a kada je završio

U tablici *tblJobStatus* u bazi je moguće pratiti status proizvodnje po aktivnostima i vremenu kada je aktivnost započela. Na slici 39 u nastavku je vidljivo da se radilo o jednom poslu, koji je prošao kroz niz koraka, status tih koraka, npr., početak zagrijavanja printera te kraj zagrijavanja printera i vremena kada je svaki korak započeo. Tako kažemo da je zagrijavanje printera započelo u 11:48, a završilo je u 12:08.

tblJobStatus

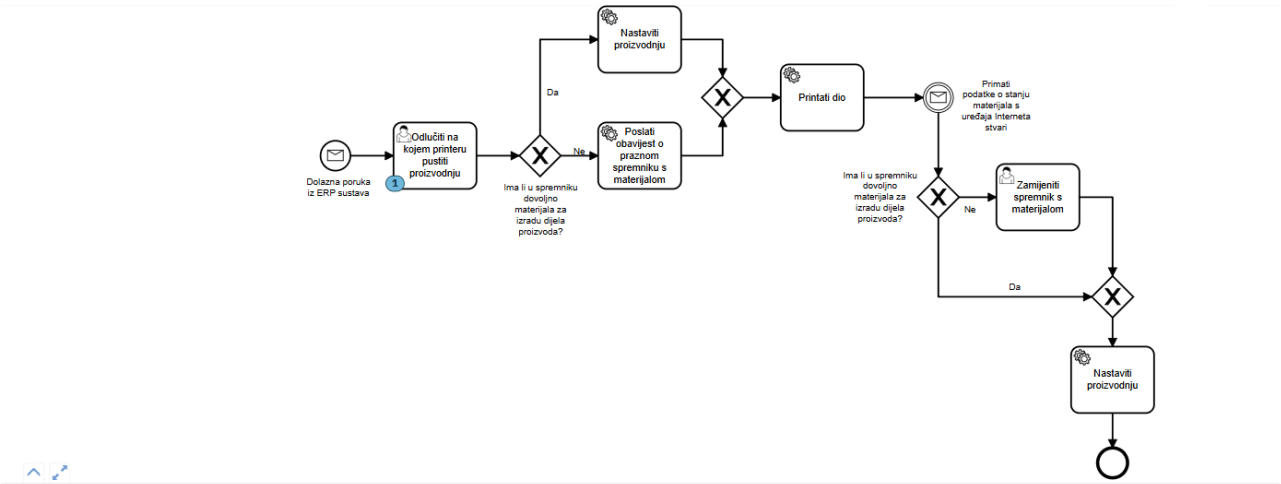
Id	Job_Id	Step	Status	Created	Click to Add
1	1	Job	Start	21/02/2025 11:48:17	
2	1	Warm Up	Start	21/02/2025 11:48:20	
3	1	Warm Up	End	21/02/2025 12:08:21	
4	1	Print	Start	21/02/2025 12:08:25	
5	1	Print	End	21/02/2025 13:38:26	
6	1	Job	End	21/02/2025 13:38:29	
(New)					

Slika 39 Status proizvodnje po aktivnostima i vremenu odvijanja

14.2 Digitalni blizanac s informacijom o količini materijala u spremniku

Na slici 40 u nastavku se nalazi digitalni blizanac proizvodnje dronova koji upozorava na količinu materijala u spremniku te je to, prema predloženoj metodici za razvoj proizvodnih digitalnih blizanaca, aktivnost 4 u kojoj se razvija proizvodni digitalni blizanac. Ovo je vrlo jednostavan primjer prognoziranja kada će u spremniku s materijalom za printanje nedostajati materijala za izradu dijela proizvoda. Količina materijala u spremniku 3D pisača nadzire se putem mjernog sustava temeljenog na laseru. Senzor je postavljen okomito prema roli spremnika i detektira je li udaljenost mala (kada nit prolazi kroz područje usmjerenog lasera) ili veća (kada se zraka odbija od unutarnje stijenke kućišta printera). Laser se usmjerava prema objektu, a reflektirano svjetlo prolazi kroz drugu leću unutar senzora. Ovisno o udaljenosti, mijenja se kut refleksije, što omogućuje senzoru da triangulacijom precizno izračuna stvarnu udaljenost. Analizom razlike između očekivane i izmjerene razine u milimetrima, sustav otkriva postoji li nedovoljna količina materijala za dovršetak ispisa. Ako ima dovoljno materijala, proizvodnja se nastavlja, ako će uskoro nedostajati materijala, šalje se obavijest putem upravljačke ploče Node-RED-a i proizvodnja se nastavlja, odnosno, dio se printa. Digitalni blizanac, nakon što je dio isprintan, prima podatke o stanju materijala sa senzora te se ponovno ispituje da li u spremniku ima dovoljno materijala za izradu dijela proizvoda (aktivnost 4.2. predložene metodike gdje se uključuje komunikacijski element prema vanjskom sustavu). Ako ima, spremaju se podaci o količini materijala u spremniku te se proizvodnja nastavlja. Ako materijala nema dovoljno, senzor na vratima printera svijetli crveno što je znak radniku da zamijeni spremnik s materijalom. Nakon što radnik zamijeni spremnik s materijalom, nova proizvodnja na tom printeru može početi.

Na slici 40 u nastavku se nalazi prikaz pokrenutog proizvodnog digitalnog blizanca u Camunda platformi i to predstavlja aktivnost 5 predložene metodike gdje se digitalni blizanac pušta u pogon. Pošto se proizvodi jedan po jedan dio, tako je i u sustav ušla jedna značka, odnosno, jedan *token*. Kako teče proizvodnja, tako se mijenja putanja kojom značka prolazi kroz sustav.

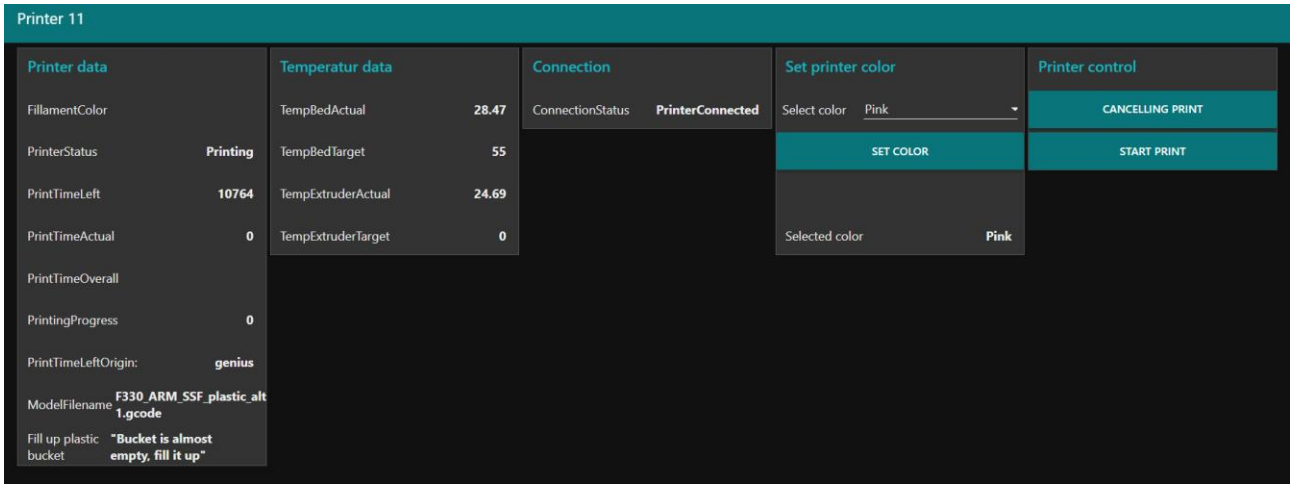


Variables | Incidents | Called Process Instances | User Tasks | Jobs | External Tasks

Activity	Assignee	Owner	Creation Date	Due Date	Follow Up Date	Priority	Delegation State	Task ID
Odlučiti na kojem printeru pustiti proizvodnju						50		e94a67dd-b0a3-11ef-b85d-c01850961d7e

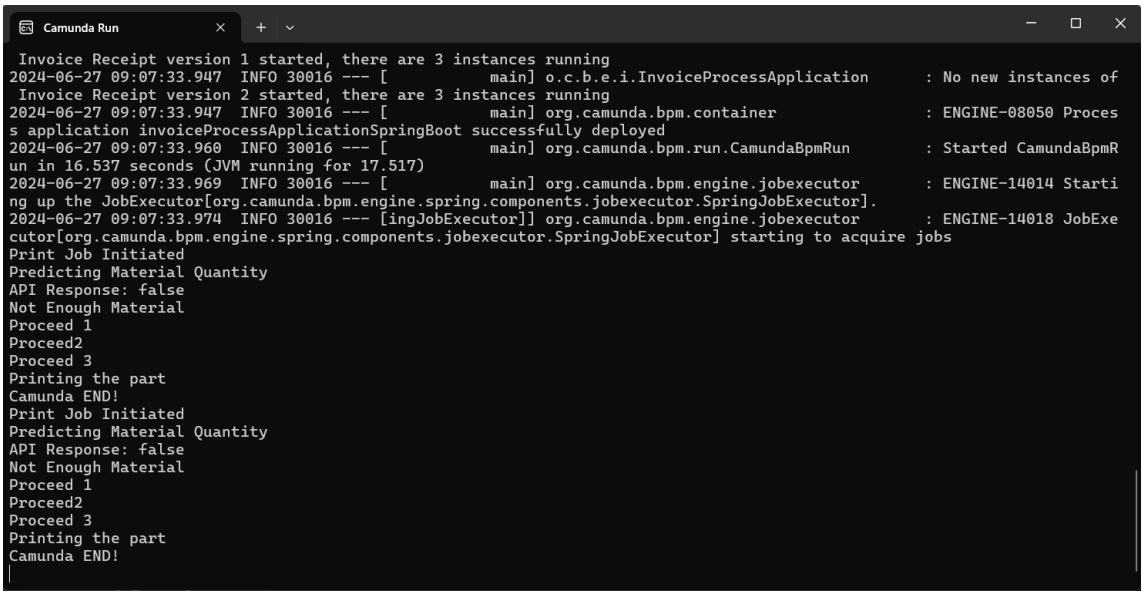
Slika 40 Pokrenut proizvodni digitalni blizanac za proizvodnju dronova u Camundi

Kako bi se prikazalo da proizvodni digitalni blizanac komunicira s vanjskim svijetom i šalje informacije o količini materijala u spremniku, priložena je slika 41 u nastavku koja prikazuje upravljačku ploču u *Node-Redu* nakon što je proces proizvodnje pokrenut. Ako će uskoro ponestati materijala za izradu dijela proizvoda, na upravljačkoj ploči se javlja poruka „*Bucket is almost empty. Fill it up*“, što u prijevodu znači „Spremnik je skoro prazan. Napuni ga“.



Slika 41 Upravljačka ploča printera iz NodeReda nakon pokretanja procesa

U konzoli je također moguće pratiti status proizvodnje. Primjer toga je slika 42 u nastavku na kojoj je prikazano kako platforma Camunde također prima podatke o proizvodnji i nedostatku materijala potrebnog za proizvodnju. Kako proizvodnja teče, tako konzola prima podatke i radnik može pratiti stanje materijala u spremniku. Ovo je aktivnost 6 predložene metodike gdje se prati izvođenje proizvodnog digitalnog blizanca.



Slika 42 Camunda prima podatke o stanju izvođenja procesa

14.3 Digitalni blizanac proizvodnje mobilnih uređaja

Kao što je već ranije istaknuto, prediktivno održavanje opreme u kontekstu razvoja proizvodnih digitalnih blizanaca predstavlja značajan element kako bi se osigurala neprekidnost proizvodnje i optimizacija rada opreme. U okviru pametne proizvodne linije s FSB-a, koja simulira sklapanje, odnosno, proizvodnju mobilnih uređaja, razvijen je sustav prediktivnog održavanja za dva ključna stroja: bušilicu i prešu. Cilj je predvidjeti potencijalne kvarove prije nego što dođe do zastoja u proizvodnom procesu, čime se minimiziraju troškovi održavanja i potencijalni gubici u proizvodnji. Prognozirani kvarovi obuhvaćaju pet tipičnih, već ranije navedenih, kategorija: (1) kvar uslijed trošenja alata, koji je posljedica dugotrajnog korištenja reznih i obradnih alata; (2) kvar uslijed disipacije topline, pri čemu dolazi do pregrijavanja komponenti; (3) kvar uslijed napajanja, koji se odnosi na nestabilnosti u električnom napajanju; (4) kvar uslijed preopterećenja, koji nastaje zbog prekoračenja kapaciteta opterećenja opreme; te (5) nasumični kvarovi, koji se pojavljuju s vjerojatnošću od 0,1 % neovisno o uvjetima proizvodnog procesa. Budući da je u realnim uvjetima rada teško prikupiti podatke koji bi opisivali rad i stanje opreme, za potrebe razvoja prediktivnog modela korišten je sintetički skup podataka preuzet o radu strojeva. Ti podaci obuhvaćaju sljedeće parametre: temperaturu zraka, temperaturu procesa, moment zakretanja i trošenje alata. Na temelju analize tih parametara razvijen je prediktivni model koji omogućuje pravovremenu detekciju potencijalnih kvarova i proaktivno održavanje opreme. Detaljan opis podataka je naveden ranije u radu u potpoglavlju *Podaci korišteni za slučajeve prediktivnog održavanja strojeva*.

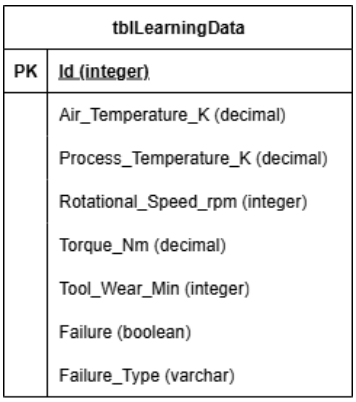
Za potrebe izrade digitalnog blizanca proizvodnje mobilnih uređaja, napravljena je nova relacijska baza podataka u Microsoft Accessu. Izrada relacijske baze podataka provedena je s ciljem objedinjavanja stvarnih podataka preuzetih s FSB-a, na kojem se nalazi školska proizvodna linija, i sintetičkih podataka. Ova baza podataka služi kao temelj za razvoj prediktivnog održavanja opreme u okviru digitalnog blizanca, koji simulira proizvodnju mobilnih uređaja.

Polazište za izradu baze podataka je bila arhivska baza podataka FSB-a, koja sadrži povijesne podatke o kretanju proizvodnje i korištenju opreme, odnosno, resursa, na stvarnoj proizvodnoj liniji. Analizom

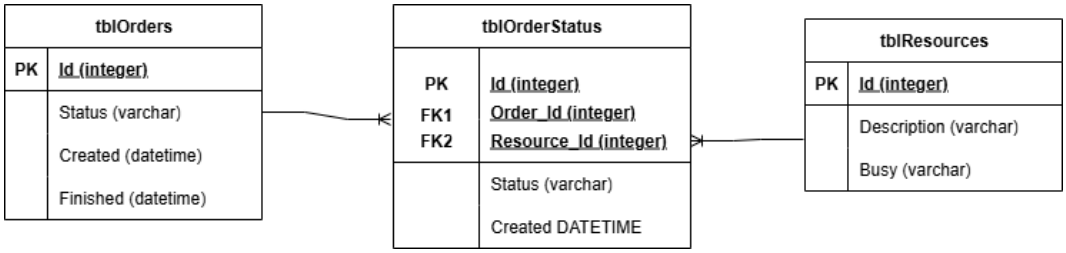
sadržaja te baze dobiven je uvid u stvarni poslovni proces i strukturu podataka koji prate rad proizvodne linije. Za potrebe istraživanja, struktura baze podataka je pojednostavljena i prilagođena na način da sadrži samo elemente koji su potrebni za rad digitalnog blizanca, odnosno, koji se tiču samog tijeka poslovnog procesa.
Za punjenje baze podataka korištena je Python skripta iz Priloga 5 ovog dokumenta. Njezina svrha bila je automatizacija procesa kreiranja tablica i unosa podataka u Microsoft Access bazu podataka. Skripta je izrađena s ciljem povezivanja dvaju izvora podataka: podataka o uvjetima rada strojeva preuzetih iz otvorenog skupa podataka te stvarnih podataka s proizvodne linije na FSB-u.
Python skripta za izradu baze sastoji se od nekoliko koraka i funkcionalnih cjelina:

- Postavljanje radnog direktorija: Skripta započinje postavljanjem radnog direktorija pomoću funkcije *set_working_directory(path)* kako bi se osiguralo da se svi potrebni ulazni i izlazni podaci nalaze u predviđenoj mapi.
- Uspostavljanje veze s bazom podataka: Funkcija *connect_to_database(db_path)* koristi biblioteku *pyodbc* za spajanje na Microsoft Access bazu podataka. Prilikom uspostavljanja veze koristi se ODBC upravljački program za Access baze, a kao parametar se navodi putanja do datoteke baze podataka koja je prije toga ručno kreirana.
- Brisanje postojećih tablica: Funkcija *drop_table(cursor, table_name)* omogućava brisanje postojećih tablica u bazi podataka. Ako tablica s određenim imenom već postoji, bit će obrisana kako bi se izbjegli konflikti prilikom kreiranja novih tablica s istim nazivima. Ako tablica ne postoji, funkcija ispisuje poruku i nastavlja s izvršavanjem.
- Kreiranje novih tablica: Funkcija *create_table(cursor, table_name, schema)* kreira tablice u bazi podataka prema unaprijed definiranim shemama. Sheme tablica specificiraju nazive stupaca, njihove tipove podataka (npr. SINGLE, TEXT, DATETIME), te primarne ključeve. Ključne tablice kreirane ovom funkcijom su:
    - *tblLearningData* – sadrži sintetičke podatke iz otvorenog skupa podataka o uvjetima rada strojeva: temperatura zraka, temperatura procesa, moment zakretanja, trošenje alata, status kvara i tip kvara. Ova tablica se koristi kao izvor podataka za učenje neuronske mreže.
    - *tblResources* – sadrži podatke o resursima, njihovim identifikatorima, opisima i statusima zauzetosti.
    - *tblOrders* – sadrži podatke o narudžbama, odnosno, statusu narudžbe, vremenu kada je narudžba inicirana i kada je završena.
    - *tblOrderStatus* – bilježi promjene statusa narudžbi i resursa tijekom vremena.

Na slikama 43 i 44 se nalaze prikaz strukture sintetičkih podataka te ERA dijagram radi prikaza strukture podataka iz realnog sustava koji prikazuje podatke o resursima i narudžbama.



Slika 43 Tablica tblLearningData



Slika 44 ERA dijagram s prikazom odnosa između resursa i narudžbi

- Učitavanje podataka iz .csv datoteka: Skripta učitava podatke iz dvije csv datoteke pomoću biblioteke *pandas*:
    - *predictive_maintenance.csv* je datoteka koja sadrži podatke o uvjetima rada strojeva, uključujući temperaturu zraka, temperaturu procesa, moment zakretanja, trošenje alata, status kvara i tip kvara.
    - *resources.csv* sadrži podatke s FSB-a o resursima u proizvodnom procesu, uključujući njihove opise i status zauzetosti.
- Mapiranje i prilagodba podataka: Budući da nazivi stupaca u CSV datotekama nisu u potpunosti odgovarali nazivu polja u Access tablicama, skripta koristi rječnike za preslikavanje naziva stupaca u željeni format (npr. *Air temperature [K]* u *Air_Temperature_K*). Također, podaci o statusu kvara (tipa *Failure*) i zauzetosti resursa (tipa *Busy*), koji su u CSV datotekama predstavljeni logičkim vrijednostima (*True/False*), konvertirani su u formate kompatibilne s Access bazom (-1 za *True* i 0 za *False*).
- Unos podataka u tablice: Funkcija *insert_data(cursor, table_name, columns, data)* koristi SQL naredbu *INSERT INTO* za upis podataka u tablice baze. Podaci se unose redak po redak, pri čemu se koriste pripremljene varijable (eng., *placeholders*) za dinamičko popunjavanje upita.
- Zatvaranje veze: Nakon što su sve tablice kreirane i popunjene podacima, skripta zatvara kursor i prekida vezu s bazom podataka kako bi se osigurala ispravna pohrana podataka.

Razlog izrade nove relacijske baze podataka leži u potrebi za objedinjavanjem podataka iz različitih izvora i stvaranjem strukturirane i pregledne podloge za razvoj digitalnog blizanca proizvodnog poslovnog procesa. Baza podataka s povijesnim podacima s FSB-a poslužila je kao podloga za razumijevanje procesa i sadržaja, dok su podaci iz otvorenog skupa omogućili dobivanje reprezentativnog skupa podataka za treniranje prediktivnog modela. Krajnji rezultat je relacijska baza

podataka, koja predstavlja osnovu za izvođenje proizvodnog digitalnog blizanca i prediktivno održavanje opreme unutar njega.

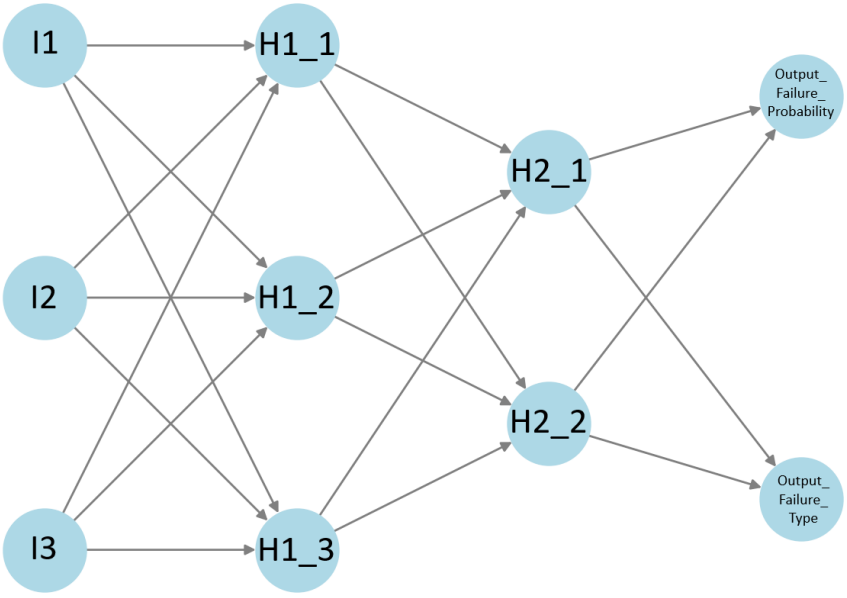
Kao i kod digitalnog blizanca proizvodnje dronova, gdje je prediktivna analitika imala ulogu pratiti stanje printera, tako je i u ovom primjeru korištena neuronska mreža kako bi se omogućilo prediktivno održavanje strojeva, točnije, bušilice i preše (aktivnost 4.1. iz predložene metodike za razvoj proizvodnih digitalnih blizanaca koja se odnosi na ugradnju složenijeg poslovnog pravila). Za tu svrhu napravljena je Python skripta iz Priloga 6 ovog dokumenta koja ima dvostruku funkciju, a služi za treniranje neuronske mreže koristeći podatke iz Access baze podataka o uvjetima rada proizvodne opreme te predviđanje kvara i vrste kvara nad opremom na temelju novih podataka ili nasumično odabranog zapisa iz baze, putem REST API-ja izgrađenog pomoću Flask aplikacije. Ovo je aktivnost 4.1.1. iz predložene metodike u kojoj se razvija prediktivni model za predviđanje održavanja strojeva. Ova skripta radi na način da učitava podatke iz Access baze, obrađuje podatke na način da se kodira kategorijska varijabla "*Failure_Type*" koristeći *LabelEncoder* (npr. kvar uslijed trošenja alata → broj 0). Zatim se podaci dijele na:

- *X*: ulazne značajke (temperatura, brzina, moment…)
- *y_failure*: binarna vrijednost (kvar ili ne)
- *y_failure_type*: tip kvara (kategorijski)

te se radi standardizacija (skaliranje) ulaznih podataka pomoću *StandardScaler* radi postizanja boljih rezultata neuronske mreže. Nakon ovoga, definirana je neuronska mreža s dvostrukim izlazom koja istovremeno predviđa vjerojatnost nastanka kvara (*Failure*) i tip kvara (*Failure_Type*). Mreža se sastoji od ulaznog sloja čija dimenzionalnost odgovara broju ulaznih značajki, nakon čega slijede dva skrivena sloja s 64, odnosno 32 neurona, pri čemu se u oba sloja koristi *ReLU (Rectified Linear Unit)* aktivacijska funkcija. Iz skrivenih slojeva mreža se grana u dva izlazna sloja, pri čemu je prvi izlaz namijenjen predikciji kvara, a drugi identifikaciji tipa kvara. Oba izlazna sloja koriste *softmax* aktivacijsku funkciju zbog višeklasnih ishoda koje predviđaju. Za treniranje modela koristi se funkcija gubitka: kategorijska unakrsna entropija (eng. *categorical_crossentropy*), dok je za optimizaciju odabran Adam optimizator, koji je poznat po efikasnosti u radu s kompleksnijim modelima i većim količinama podataka. Ovdje se radi o aktivnosti 4.1.2. iz predložene metodike kada se trenira neuronska mreža. Kao evaluacijska metrika koristi se točnost (eng. *accuracy*), čime se prati uspješnost predikcije, kako za nastanak kvara, tako i za njegov tip. Model se trenira u trajanju od 50 epoha s veličinom paketa podataka (eng. *batch size*) od 32, čime se osigurava stabilno konvergiranje mreže i zadovoljavajuća kvaliteta predikcija. Radi se o jednoj višeslojnoj perceptronskoj mreži za klasifikacijske zadatke, s dvostrukim izlazom, što je česta arhitektura kada se predviđaju dvije povezane ciljne varijable. U ovom slučaju, mreža predviđa:

- hoće li doći do kvara (binarna klasifikacija) i
- koja vrsta kvara će nastupiti (višeklasna klasifikacija).

Prikaz neuronske mreže s dvostrukim izlazom se nalazi na slici 45 u nastavku.



Slika 45 Neuronska mreža s dvostrukim izlazom

Ova višezadaćna (eng. *multi-output*) struktura neuronske mreže je prikladna za ovaj slučaj upotrebe jer mreža istovremeno uči oba zadatka, pri čemu poboljšava točnost jer uči korelacije između nastanka kvara i tipa kvara.

Python skripta se koristi i za izradu Flask API aplikacije za predikcije gdje su definirane rute za:

- /*drill_status* – za predikciju statusa bušilice.
- /*press_status* – za predikciju statusa preše.

Senzorski podaci se simuliraju očitavanjem podataka sa stroja tako što se nasumično odabire jedan zapis iz baze podataka. Nakon odabira, podaci se pripremaju i skaliraju prema istim parametrima koji su korišteni prilikom treniranja neuronske mreže, kako bi ulazni podaci bili u istom rasponu vrijednosti. Učitava se prethodno trenirani model, koji na temelju ulaznih parametara predviđa vjerojatnost nastanka kvara, primjerice 23,4%, te određuje tip mogućeg kvara, poput „Trošenja alata“. Sada se dolazi do aktivnosti 4.1.3. iz predložene metodike kada se prediktivni model održavanja ugrađuje u proizvodni digitalni blizanac. Dobiveni rezultati predikcije vraćaju se kao JSON odgovor, što omogućuje daljnju obradu i integraciju u sustav digitalnog blizanca. Ako je vjerojatnost kvara (eng., *Failure_probability*) > 50%, onda se specificira koji je kvar, inače je rezultat da kvara nema (eng., *Failure none*).

Razlika između Python skripte iz Priloga 1 i Python skripte iz Priloga 6 je ta što se u Prilogu 1 kod koda za održavanje 3D printera koriste dvije odvojene neuronske mreže. Dakle, za razliku od koda kod prediktivnog održavanja bušilice i preše iz Priloga 6, gdje je korištena jedna neuronska mreža s dvostrukim izlazom, ovdje su definirane dvije potpuno zasebne mreže, svaka za svoj zadatak.

Prva mreža je binarna klasifikacijska neuronska mreža i koristi se za predviđanje hoće li doći do kvara opreme (*Failure*/*No Failure*). Druga mreža je višeklasna klasifikacijska neuronska mreža i koristi se za predviđanje tipa kvara ako kvar nastupi. Ova mreža ima više izlaznih neurona, ovisno o broju različitih tipova kvarova. Dakle, kod održavanja printera se radilo o dvije odvojene neuronske mreže, dok se kod održavanja bušilice i preše radilo o jednoj mreži s dva izlaza. Oba pristupa su ispravna i često se koriste u praksi, s time da višezadaćne mreže (poput ove za održavanje bušilice i preše) imaju prednost jer istovremeno uče iz međusobne povezanosti dvaju izlaza, dok dvije odvojene mreže omogućuju veću fleksibilnost, ali svaka uči neovisno. Cilj istraživanja je bio isprobati oba pristupa

jer je namjera bila prikazati različite načine implementacije neuronskih mreža za rješavanje klasifikacijskih zadataka u prediktivnom održavanju. Time se prikazalo kako se mogu koristiti različite arhitekture višeslojnih perceptronskih mreža koje, unatoč sličnim osnovama, nude fleksibilnost u pristupu izradi modela. Cilj je bio naglasiti da postoji više prihvatljivih puteva do rješenja, ovisno o specifičnim zahtjevima i preferencijama implementacije.
Oba pristupa, odnosno, mreže iz spomenutih slučajeva upotrebe spadaju u višeslojne perceptronske mreže (MLP), koje su vrsta *feedforward* neuronskih mreža. Koriste gusto povezane (eng., *fully connected*) slojeve s *ReLU* aktivacijskom funkcijom u skrivenim slojevima, što je česta arhitektura za klasifikacijske zadatke u prediktivnom održavanju te se stoga smatraju pametnim odabirom za slučajeve korištenja prikazane u ovom istraživanju. Da obje implementacije koriste višeslojne perceptronske mreže (MLP) se vidi kroz sekvencijalne modele koji koriste *Dense* slojeve. Obje mreže su *feedforward*, a to znači da se informacije prenose samo prema naprijed, od ulaznog prema izlaznom sloju. Nadalje, koriste se gusto povezani slojevi, implementirani kroz *Dense* slojeve u TensorFlow/Keras-u te je svaki neuron povezan sa svim neuronima prethodnog sloja. I zadnje, *ReLU* aktivacijska funkcija se koristi u skrivenim slojevima, a to znači da u oba koda vidimo *activation='relu'* u skrivenim slojevima i za izlazne slojeve koriste se prikladne aktivacijske funkcije za klasifikaciju (*sigmoid* za binarnu i *softmax* za višeklasnu klasifikaciju).
Ovo je česti pristup izrade arhitekture za zadatke prediktivnog održavanja iz nekoliko razloga, a oni su:

- MLP mreže su učinkovite u prepoznavanju složenih obrazaca u numeričkim podacima sa senzora,
- *ReLU* aktivacijska funkcija pomaže u rješavanju problema nestajućeg gradijenta[10] i omogućuje učinkovito učenje,
- arhitektura je dovoljno jednostavna za implementaciju, ali i dovoljno učinkovita za predviđanje kvarova opreme kod proizvodnog digitalnog blizanca.

S ciljem prikaza izvođenja proizvodnog digitalnog blizanca stvarnog procesa na pametnoj proizvodnoj liniji FSB-a, razvijena je Python skripta pod nazivom *mes_simulator.py* iz Priloga 7 ovog dokumenta. Navedena skripta se koristi za simulaciju stvarnog tijeka proizvodnje u kontroliranom digitalnom okruženju te omogućuje generiranje podataka u svrhu testiranja i unapređenja digitalnog blizanca proizvodnog poslovnog procesa. Proces simulacije temelji se na stvarnim podacima prikupljenima s FSB-a, a skripta je oblikovana tako da što vjernije imitira redoslijed i trajanje aktivnosti koje se odvijaju u realnom proizvodnom procesu. Krajnji rezultat rada simulatora je zapisivanje svake simulirane narudžbe i njezinog statusa u relacijsku bazu podataka Access, čime se osigurava konzistentnost podataka s onima dobivenima u stvarnom proizvodnom okruženju. Znači, *mes_simulator* je izrađen na temelju stvarnih podataka prikupljenih iz proizvodnje na FSB-u i reproducira tipične tijekove aktivnosti te trajanja pojedinih proizvodnih koraka. Generirani podaci koriste se za validaciju i razvoj digitalnog blizanca.

[10] Tijekom treniranja neuronske mreže koristi se metoda propagacije unazad (eng., *backpropagation*) za ažuriranje težina. Ova metoda računa gradijent greške s obzirom na težine, krećući se od izlaza mreže prema ulazu (od stražnjih slojeva prema prednjima). Ako se koriste aktivacijske funkcije poput *sigmoid* ili *tanh*, derivacije tih funkcija u određenim područjima su jako male (blizu 0). Kada se gradijenti množe tijekom propagacije unazad, što je mreža dublja, gradijenti postaju sve manji te se kaže da "nestaju". To znači da slojevi bliže ulazu u mrežu dobivaju vrlo male promjene težina, što usporava ili potpuno zaustavlja učenje. Zato *ReLU* aktivacijska funkcija pomaže jer kada je ulaz pozitivan, gradijent je 1, što sprječava nestajanje gradijenta te brže konvergira kod dubokih mreža u usporedbi sa *sigmoid* ili *tanh*.

Skripta koristi asinkroni okvir za izvršavanje simulacije više narudžbi istovremeno, pri čemu svaka narudžba prolazi kroz zadani niz koraka definiranih u varijabli *PROCESS_STEPS*. Ti koraci uključuju preuzimanje dijela iz spremnika (eng., *magazine*), premještanje do bušilice (eng., *transit to drill*), obradu na bušilici (eng., *drill*), transport do mobilnog robota (eng., *transit to Robotino*), rad mobilnog robota (*Robotino*), vraćanje robota na početni položaj nakon ručne montaže, odnosno, ljudske aktivnosti (eng., *Robotino return*), transport do preše (eng., *transit to press*), primjenu preše (eng., *press*) te, konačno, premještanje gotovog proizvoda za preuzimanje (eng., *transit to collection*). Svaka radna stanica ima dodijeljene vremenske intervale trajanja aktivnosti, definirane u varijabli *RESOURCE_INTERVALS*, koji su postavljeni na temelju opažanja u realnom proizvodnom procesu. U skripti se isto tako može vidjeti simulacija ljudske aktivnosti na radnoj stanici, gdje radnik ručno spaja dijelove mobilnog uređaja (*front cover* i *back cover*). Ova faza, u kojoj se proizvod sklapa na radnoj stanici od strane radnika pa se stavlja ponovno na Robotina, zahtijeva ručnu potvrdu završetka aktivnosti od strane radnika na Robotinu i u Camundi, što ćemo vidjeti u nastavku. To je u skripti riješeno kroz *manual_task_finished*, koji simulira da je ručna aktivnost završena, nakon čega Robotino može nastaviti kretanje nazad prema proizvodnoj liniji. Ova funkcionalnost odražava stvarnu situaciju na FSB-u, gdje radnik fizički aktivira robota nakon završetka montaže, što prikazuje interakciju čovjeka i stroja u procesu.
Osim simulacije proizvodnih aktivnosti, skripta bilježi sve ključne informacije o svakoj narudžbi i resursu u Access relacijsku bazu podataka putem funkcija kao što su *register_order_start*, *register_order_status* i *register_order_end*. Ove funkcije zapisuju vrijeme pokretanja narudžbe, početak i završetak svake aktivnosti te finalizaciju narudžbi. Pritom se za svaku radnu stanicu generira zapis s vremenskom oznakom, identifikatorom narudžbe i statusom aktivnosti (START ili END), čime se stvara detaljna kronologija svakog simuliranog proizvodnog procesa, potpuno u skladu s podacima iz realnog sustava na FSB-u.
Važno je naglasiti da je *mes_simulator* razvijen kako bi omogućio više testnih izvođenja proizvodnog procesa bez potrebe za fizičkim pokretanjem opreme na FSB-u. Time je omogućeno prikupljanje većeg broja podataka pod kontroliranim uvjetima, što je ključno za provjeru ispravnosti digitalnog blizanca i razvoj prediktivnog modela za održavanje opreme. Ovaj proizvodni digitalni blizanac generira podatke prema stvarnom obrascu (temeljenom na stvarnim podacima i vremenskim intervalima). Simulirani podaci generirani ovim pristupom koriste se za analizu performansi proizvodnog procesa te kao ulaz za treniranje neuronske mreže u svrhu prediktivnog održavanja bušilice i preše. Na taj način, *mes_simulator.py* je važan zbog integracije stvarnih podataka i simuliranog poslovnog procesa unutar digitalnog blizanca, zbog čega se stvara pouzdano rješenje za unapređenje proizvodne učinkovitosti i smanjenje zastoja u radu.
Python skripta pod nazivom *mes_listener.py* iz Priloga 8 ovog dokumenta se koristi za uspostavljanje veze između proizvodnog procesa, odnosno podataka o statusu proizvodnje pohranjenih u Access bazi podataka, i proizvodnog digitalnog blizanca razvijenog u BPMN notaciji u Camundi što je aktivnost 4.2.2. iz predložene metodike koja označava izradu Python skripte za komunikaciju vanjskog sustava s digitalnim blizancem. Ova skripta služi kao posrednik između baze podataka i proizvodnog procesa, omogućujući automatsko ažuriranje stanja digitalnog blizanca na temelju promjena u proizvodnom procesu. Skripta se pokreće nakon izvršavanja simulacije proizvodnog procesa skriptom *mes_simulator.py*, koja puni bazu podataka *mes.accdb* sa statusima resursa i narudžbi. Zadatak skripte *mes_listener.py* je da te statuse kontinuirano čita iz tablice *tblOrderStatus* i prosljeđuje ih Camundi kako bi digitalni blizanac bio u svakom trenutku osvježen i sinkroniziran sa stvarnim stanjem.

Skripta koristi asinkroni okvir (eng. *asyncio*) za istovremeno obavljanje više zadataka, što omogućuje kontinuirano praćenje baze podataka i slanje poruka Camundi bez prekida. Na početku skripte definirana su tri ključna parametra:

- *DB_PATH* – putanja do Access baze podataka *mes.accdb* u kojoj se nalaze podaci o statusima narudžbi i korištenju resursa.
- *CAMUNDA_URL* – URL Camunda REST API-ja (http://localhost:8080/engine-rest/message), putem kojeg se vrši komunikacija s BPMN procesom u Camundi.
- *POLL_INTERVAL* – vremenski interval (u sekundama) između uzastopnih provjera baze podataka. U ovom slučaju postavljen je na 1 sekundu, što znači da se svakih 1 sekundu provjerava ima li novih promjena u tablici *tblOrderStatus*.

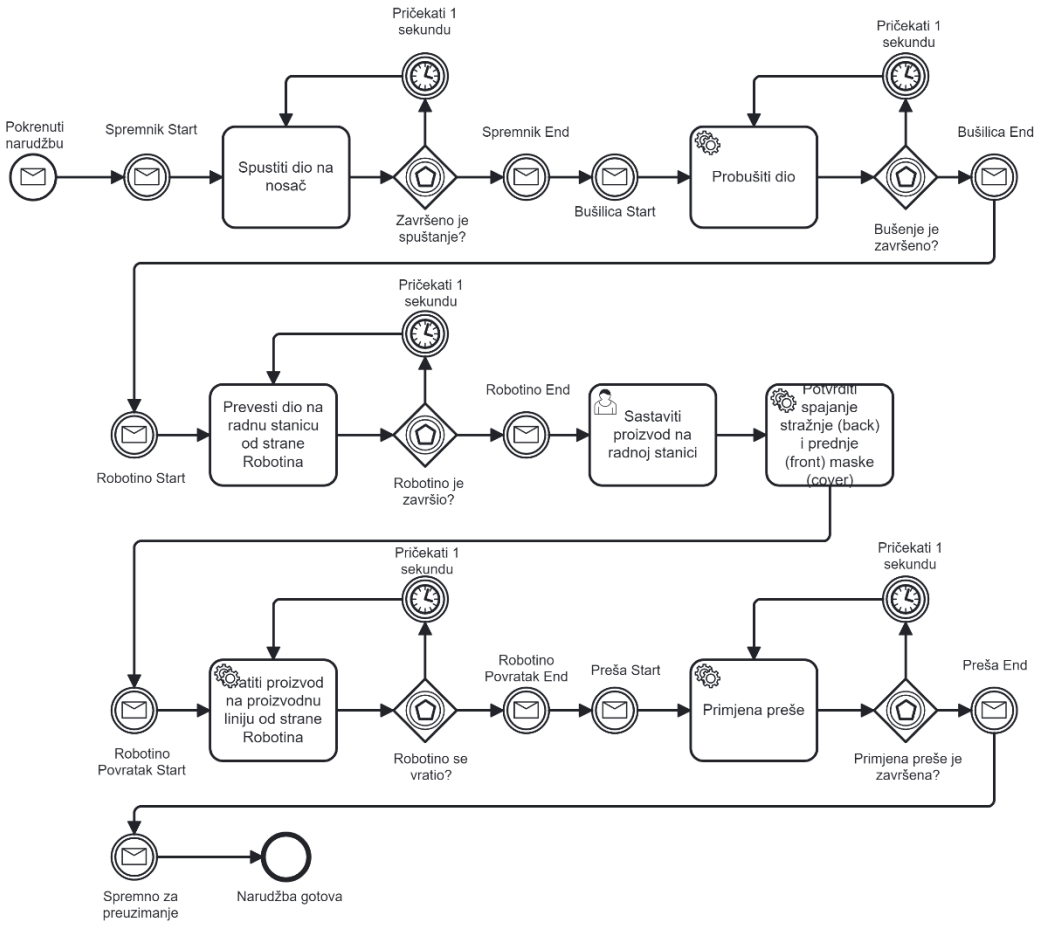
Skripta koristi skup *processed_status_ids* za praćenje već obrađenih zapisa iz tablice *tblOrderStatus*, čime se osigurava da se svaka promjena statusa pošalje Camundi samo jednom. Svakih 1 sekundu, funkcija *status_monitor()* izvršava SQL upit kojim dohvaća sve zapise iz tablice *tblOrderStatus*, sortirane prema vremenu nastanka. Skripta zatim provjerava svaki zapis i uspoređuje njegov ID s onima koji su već obrađeni. Ako naiđe na novi zapis, generira se naziv poruke na temelju resursa i statusa (npr. "*Drill Start*", "*Press End*") te se ta poruka stavlja u red obavijesti (eng. *notification_queue*), zajedno s ID-em narudžbe.

U istom trenutku, u pozadini radi funkcija *notify_camunda_worker()*, koja obrađuje taj red obavijesti i šalje REST API zahtjeve Camundi pomoću funkcije *notify_camunda()*. Ova funkcija formira JSON zahtjev u kojem se prosljeđuje naziv poruke (eng., *messageName*), kao i korelacijski ključ *orderId* ako se radi o statusima korištenja resursa. U slučaju početka narudžbe (eng., *Order Start*), šalje se procesna varijabla *orderId* kako bi se pokrenula nova instanca procesa u Camundi. Ako se radi o završetku narudžbe (eng., *Order End*), poruka se zamjenjuje s "*Ready To Collect*", što znači da je narudžba spremna za preuzimanje, odnosno, skladištenje. Svaki zahtjev šalje se *HTTP POST* metodom na *CAMUNDA_URL,* uz *Content-Type: application/json* u zaglavlju.

Skripta *mes_listener.py* je važna zato što ona omogućava funkcioniranje proizvodnog digitalnog blizanca u Camundi, jer omogućuje komunikaciju između baze podataka i BPMN modela digitalnog blizanca. Podaci koje prethodna skripta, *mes_simulator.py*, upisuje u Access bazu samo su početna točka jer tek zahvaljujući *mes_listener.py,* ti se podaci aktivno šalju Camundi, u kojoj se potom ažurira proizvodni poslovni proces i vizualno prikazuje trenutno stanje narudžbi i korištenje resursa unutar digitalnog blizanca. Na taj način, digitalni blizanac nije statički model, već dinamički odražava stvarno odvijanje proizvodnog procesa.

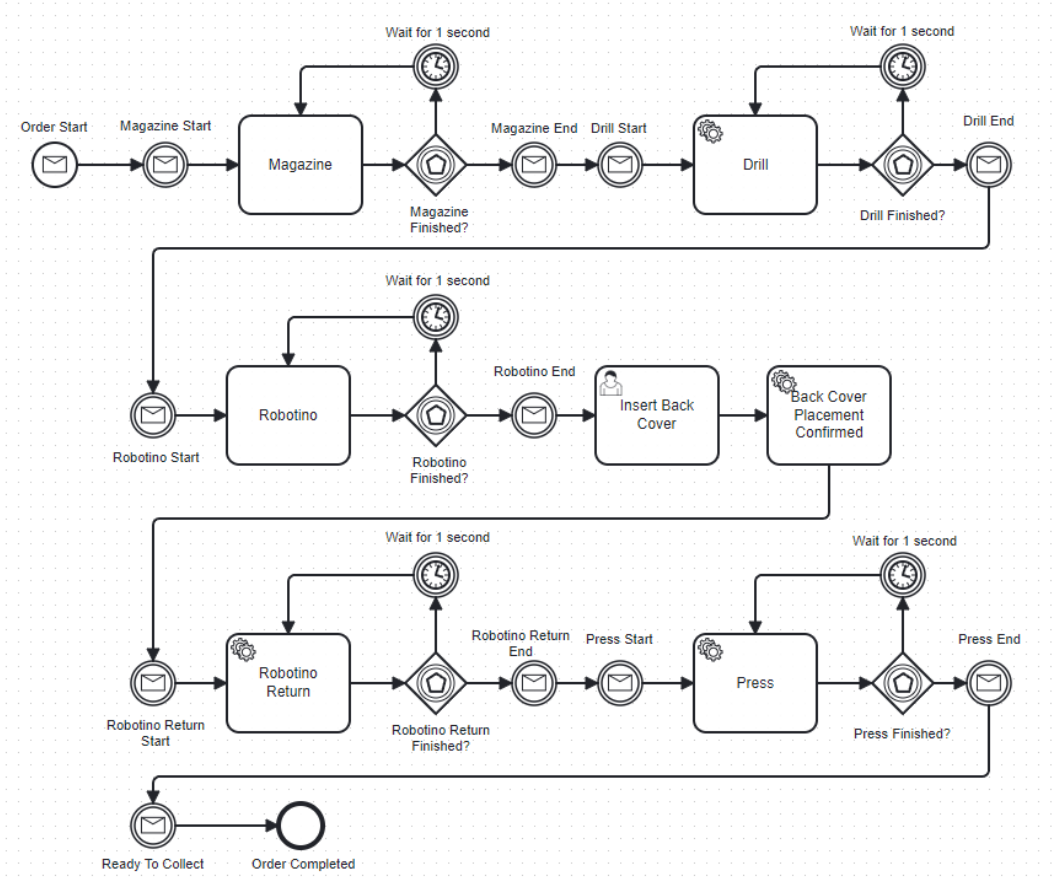
Drugim riječima, skripta *mes_listener.py* kontinuirano nadzire tablicu *tblOrderStatus* kako bi detektirala promjene statusa narudžbi te putem asinkronih zahtjeva obavještava Camundu o njihovom trenutnom stanju i poziciji u procesu. Time se omogućuje kontinuirano praćenje proizvodnog procesa u Camundi, čime se ostvaruje koncept digitalnog blizanca koji u realnom vremenu odražava promjene u proizvodnom sustavu.

Na slici 46 u nastavku se nalazi prikaz digitalnog blizanca za proizvodnju mobilnih uređaja napravljen po BPMN notaciji u Camundi te je prikaz na hrvatskom jeziku radi boljeg razumijevanja procesa. Digitalni blizanac je napravljen da odgovara realnom procesu kako se on odvija u stvarnosti te kako je to prikazano AS IS modelom ranije u istraživanju.



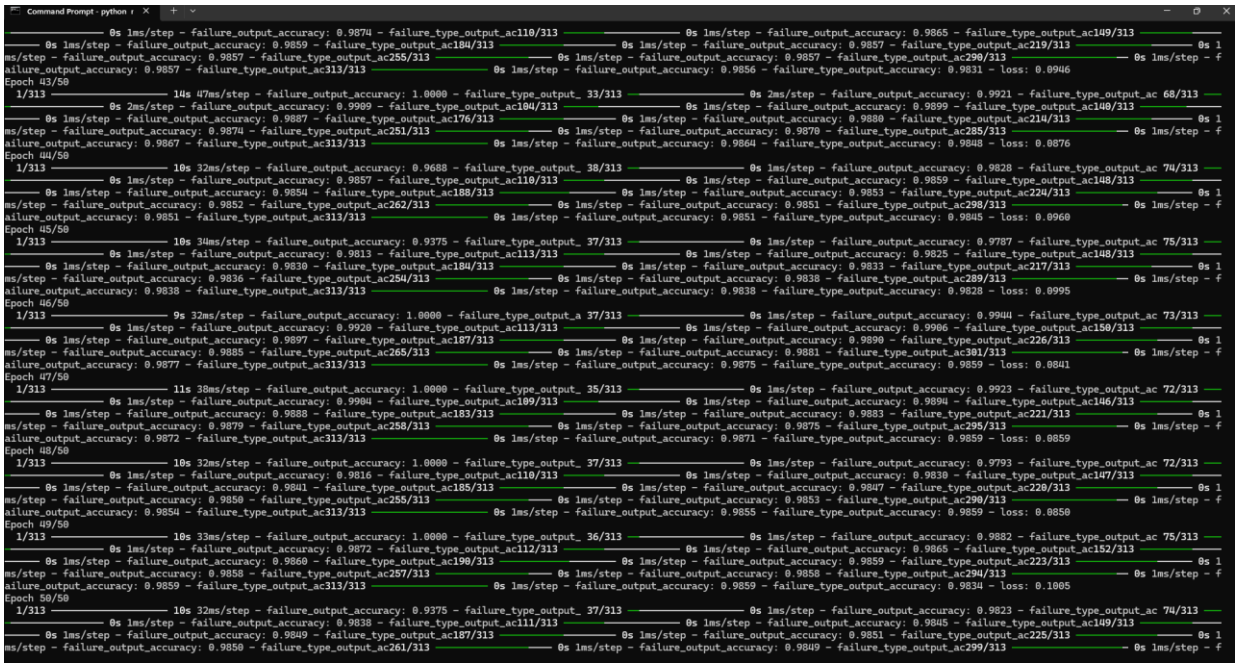
Slika 46 Digitalni blizanac proizvodnje mobilnih uređaja napravljen u Camundi, hrvatska terminologija

S obzirom na to da je programski kod u Pythonu napravljen na engleskom jeziku te je službeni jezik Camundine platforme engleski, u nastavku je prikaz digitalnog blizanca na engleskoj terminologiji. Ovaj prikaz u potpunosti translatira hrvatsku verziju digitalnog blizanca, samo je radi boljeg praćenja odvijanja procesa u konzoli, prikazana i engleska verzija modela. Razvijen proizvodni digitalni blizanac u Camundi je aktivnost 4.2.1. iz predložene metodike razvoja proizvodnog digitalnog blizanca koja se odnosi na izradu proizvodnog digitalnog blizanca u alatu za procesno servisnu aplikaciju.



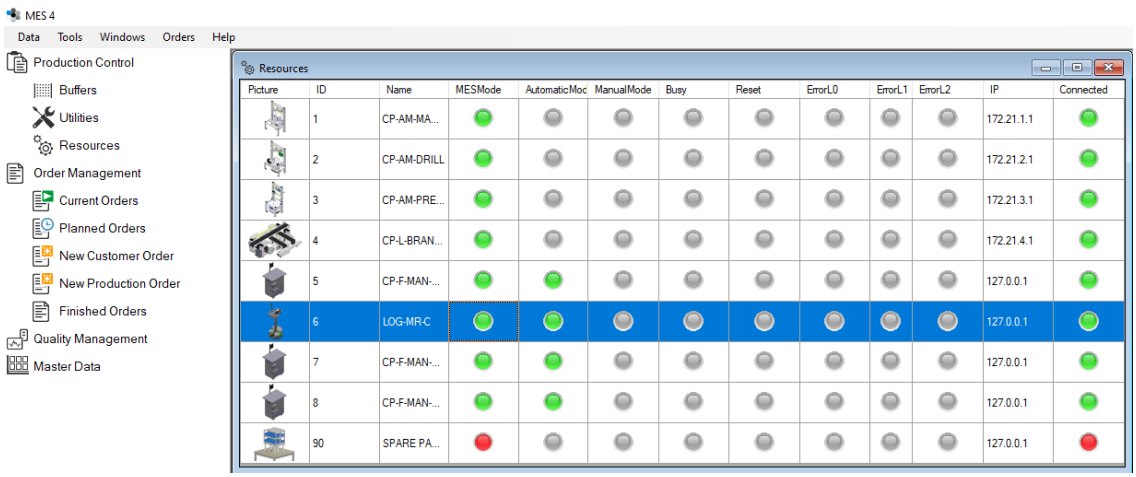
Slika 47 Digitalni blizanac proizvodnje mobilnih uređaja napravljen u Camundi, engleska terminologija

Na slici 48 u nastavku je prikazan ispis iz terminala (cmd-a) tijekom treniranja višeslojne perceptronske neuronske mreže na skupu podataka o uvjetima rada strojeva, pri čemu se prati točnost predikcije nastanka kvara (eng., *failure_output_accuracy*) i tipa kvara (eng., *failure_type_output_accuracy*) po svakoj epohi. Vidljivo je da mreža postupno postiže visoku točnost na oba izlaza, što ukazuje na uspješno treniranje modela za predikciju kvarova i identifikaciju njihovih uzroka u proizvodnom sustavu. Tijekom treniranja mreže, u svakoj epohi bilježe se metrike točnosti za dva izlaza od kojih jedan predviđa hoće li doći do kvara i drugi koji identificira tip kvara, što je karakteristično za višezadaćne neuronske mreže. Također, vidljivo je da se gubitak (eng., *loss*) kroz epohe smanjuje, dok točnost (eng., *accuracy*) na oba izlaza raste, što upućuje na stabilnu konvergenciju modela i sugerira da podaci dobro odgovaraju arhitekturi mreže.



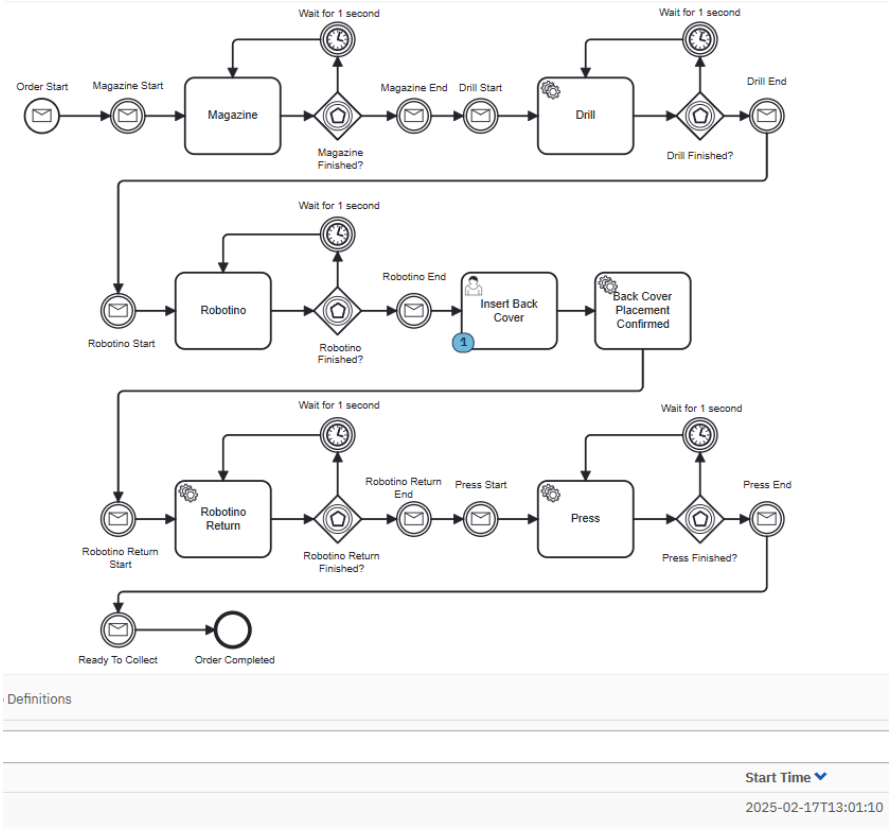
Slika 48 Prikaz uspješno trenirane neuronske mreže

Na slici 49 u nastavku je vidljivo sučelje MES sustava koje prikazuje status resursa, odnosno, strojeva u proizvodnom procesu. Na temelju dostupnih podataka, može se očitati spremnost svakog stroja za korištenje, što uključuje informacije o tome je li stroj povezan s mrežom (eng., *Connected*), je li u automatskom ili ručnom načinu rada te je li trenutno zauzet ili u stanju greške (*Error*). Zeleni indikatori signaliziraju da su strojevi spremni za rad i povezani, dok crveni indikatori upozoravaju na grešku ili nepovezanost stroja. Podaci iz MES-a o dostupnosti resursa su važni za digitalnog blizanca proizvodnog procesa jer se na temelju informacija o dostupnosti resursa može precizno upravljati proizvodnjom i prilagođavati tijek poslovnog procesa u stvarnom vremenu.



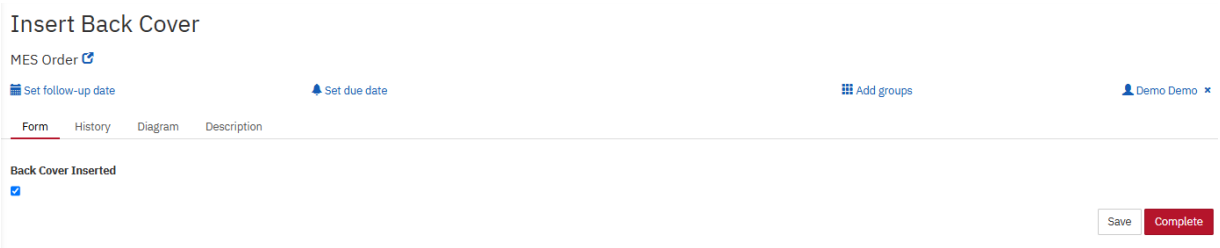
Slika 49 Praćenje spremnosti resursa iz MES 4 sustava na FSB-u

Na slici 50 u nastavku se nalazi prikaz izvođenja procesa u Camundinoj platformi odakle možemo pratiti tijek narudžbe što predstavlja aktivnost 5 predložene metodike koja uključuje puštanje proizvodnog digitalnog blizanca u pogon. Puštena je u proizvodnju prva narudžba i, prema slici, ona se nalazi na ljudskoj aktivnosti kada radnik treba sklopiti proizvod od prednjeg i stražnjeg poklopca maske. Tu započinje aktivnost 6 predložene metodike u kojoj se prati izvođenje proizvodnog digitalnog blizanca.



Slika 50 Prva narudžba u procesu na ljudskoj aktivnosti i čekanje potvrde ljudske aktivnosti

Nakon što radnik sklopi proizvod od prednje i stražnje maske, potrebno je u Camundinoj platformi popuniti generičku formu, odnosno, označiti na formi da je stražnji poklopac maske stavljen i da je ljudska aktivnost završena kako bi se proces proizvodnje mogao nastaviti. U fizičkom svijetu u ovom koraku je potrebno pritisnuti na Robotinu da je proizvod stavljen nazad na njegov nosač i da se može vratiti s proizvodom na proizvodnu liniju. Znači, ovaj dio se sastoji od nekoliko koraka: stavljanja stražnjeg poklopca na prednji (odnosno, montaže proizvoda), stavljanja proizvoda na nosač od Robotina, označavanja na formi u Camundi da je ljudska aktivnost gotova, čime se zapravo šalje i informacija u MES da je Robotino spreman za povratak.



Slika 51 Generička forma u Camundi koja prikazuje aktivnost radnika koji potvrđuje status Robotina

Slika 52 u nastavku prikazuje stanje preše u MES-u, odnosno, da je ista zauzeta proizvodnjom što predstavlja aktivnost 6.1., iz predložene metodike, koja označava prikupljanje stvarnih podataka o izvođenju digitalnog blizanca iz vanjskog sustava.

Picture	ID	Name	MESMode	AutomaticMode	ManualMode	Busy	Reset	ErrorL0	ErrorL1	ErrorL2	IP	Connected
	1	CP-AM-MAG-FR...									172.21.1.1	
	2	CP-AM-DRILL									172.21.2.1	
	3	CP-AM-PRESS									172.21.3.1	
	4	CP-L-BRANCH									172.21.4.1	
	5	CP-F-MAN-Mobile1									127.0.0.1	
	6	LOG-MR-C									127.0.0.1	
	7	CP-F-MAN-Mobile3									127.0.0.1	
	8	CP-F-MAN-Mobile2									127.0.0.1	
	90	SPARE PARTS									127.0.0.1	

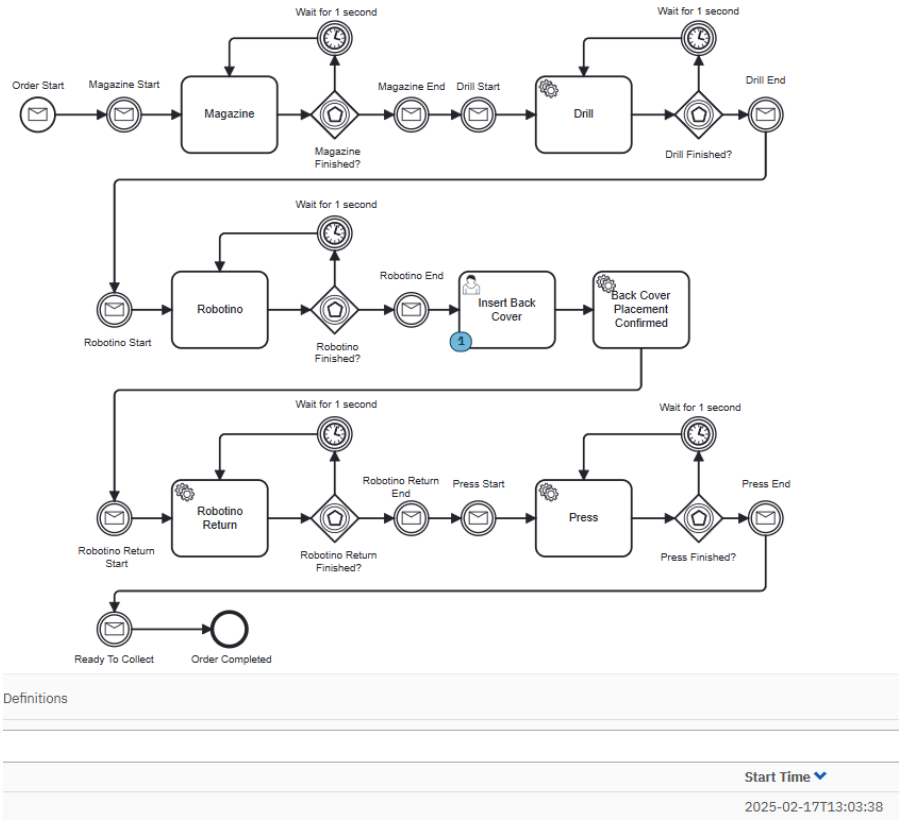
Slika 52 Prikaz zauzetosti preše

Nakon što je prva narudžba gotova, puštena je druga narudžba. Pratimo stanje strojeva preko MES sustava i u nastavu je slika 53 o zauzetosti bušilice (ponovno aktivnost 6.1.).

Picture	ID	Name	MESMode	AutomaticMode	ManualMode	Busy	Reset	ErrorL0	ErrorL1	ErrorL2	IP	Connected
	1	CP-AM-MAG-FR...									172.21.1.1	
	2	CP-AM-DRILL									172.21.2.1	
	3	CP-AM-PRESS									172.21.3.1	
	4	CP-L-BRANCH									172.21.4.1	
	5	CP-F-MAN-Mobile1									127.0.0.1	
	6	LOG-MR-C									127.0.0.1	
	7	CP-F-MAN-Mobile3									127.0.0.1	
	8	CP-F-MAN-Mobile2									127.0.0.1	
	90	SPARE PARTS									127.0.0.1	

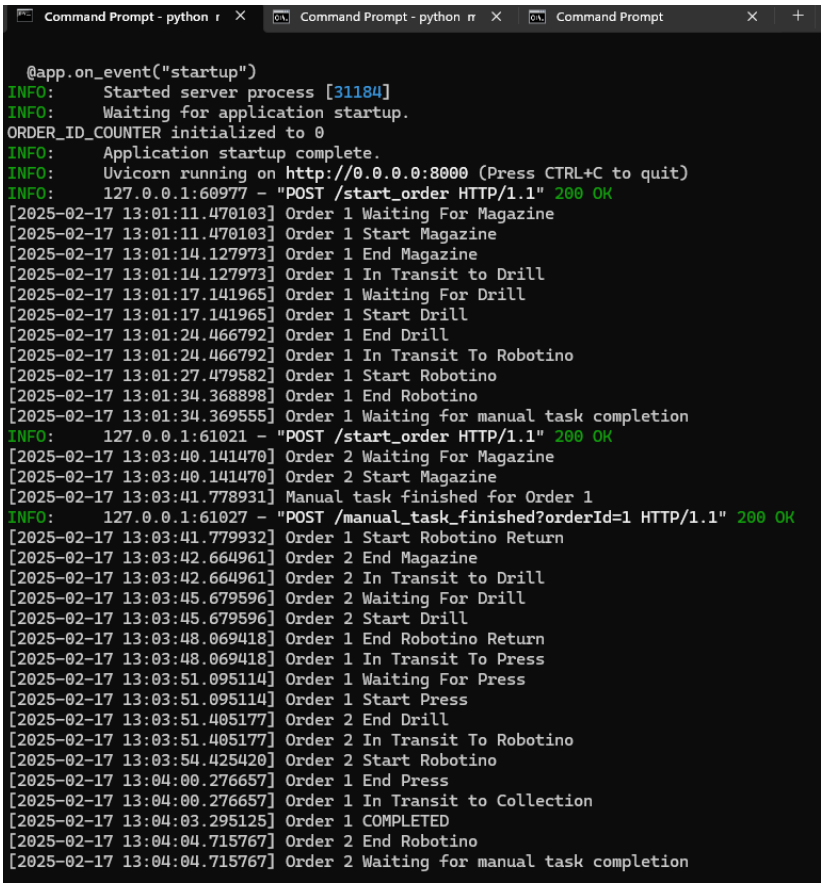
Slika 53 Prikaz zauzetosti bušilice

Druga narudžba također čeka u proizvodnom sustavu da se završi ljudska aktivnost kako bi se proizvodni proces mogao nastaviti. Slika druge narudžbe u proizvodnji nalazi se u nastavku.



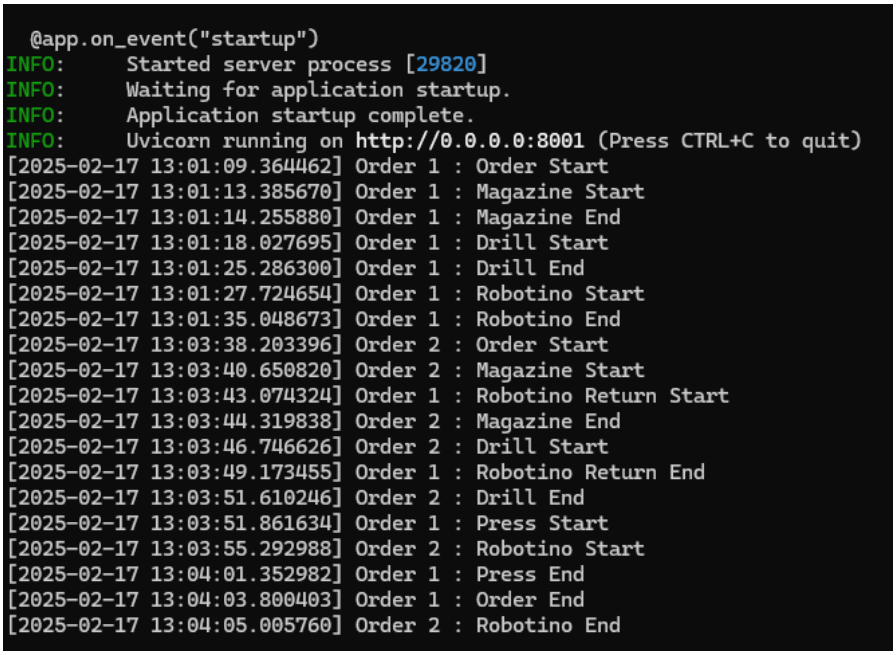
Slika 54 Provođenje druge narudžbe koja je u proizvodnom sustavu

Kako bi se pratio tijek izvođenja procesa, odnosno, proizvodnog digitalnog blizanca (aktivnost 6 iz predložene metodike), sljedeća slika 55 prikazuje terminal u kojem se vidi slijed izvođenja procesa nakon što je prva narudžba gotova, a kod druge narudžbe se čeka na završetak ljudske aktivnosti. Vidljivo je pokretanje servera, nakon čega se zaprimaju HTTP zahtjevi za pokretanje narudžbi (*POST /start_order*) te se bilježi status svake faze narudžbe. Prva narudžba (eng., *Order 1*) uspješno je završena, što potvrđuje status "*COMPLETED*", dok druga narudžba (eng., *Order 2*) čeka potvrdu radnika da je aktivnost završena (eng., *manual task completion*), što je naznačeno statusom „*Waiting for manual task completion*“. Također je vidljivo slanje HTTP zahtjeva za potvrdu ljudske aktivnosti (*POST /manual_task_finished*), što govori da se radi o integraciji s Camunda platformom.



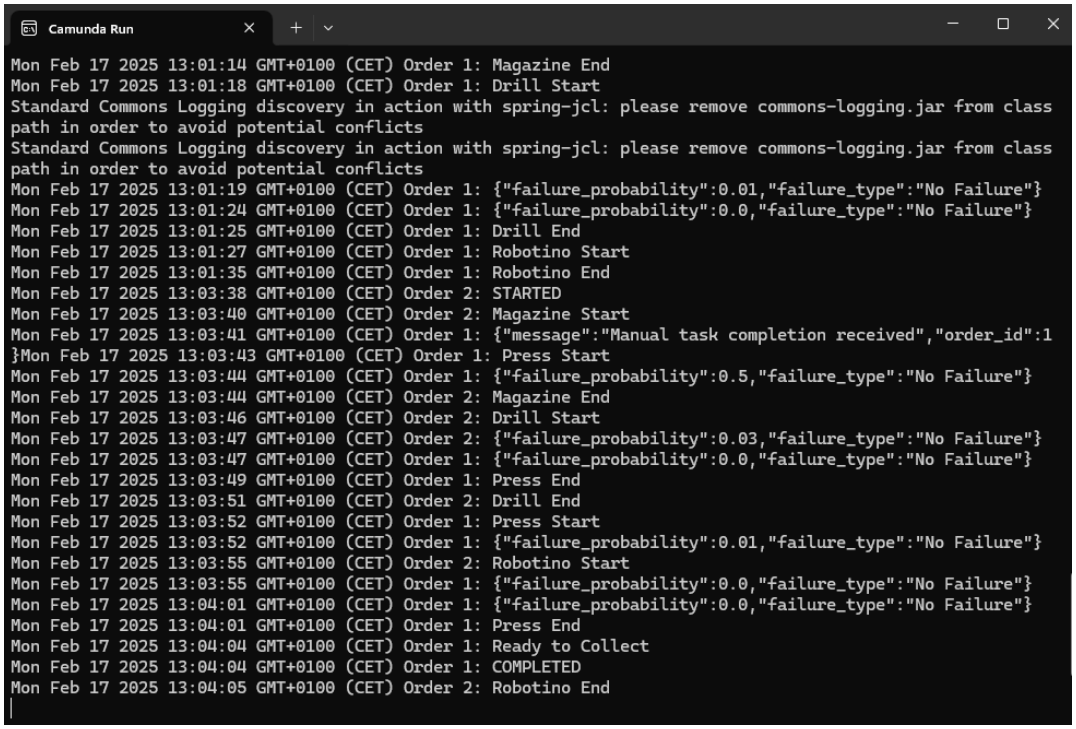
Slika 55 Tijek izvođenja procesa, prva je narudžba gotova, a druga čeka da se potvrdi forma odnosno završetak ljudske aktivnosti

Mes_listener čita stanje u bazi i obavještava Camundu gdje se proces trenutno nalazi, a prikaz navedenog se nalazi na slici 56 u nastavku. Slika 56 u nastavku je prikaz stanja procesa nakon što je prva narudžba gotova, a druga narudžba čeka da se potvrdi završetak ljudske aktivnosti u Camundi i na Robotinu kako bi se mogao vratiti na proizvodnu liniju.



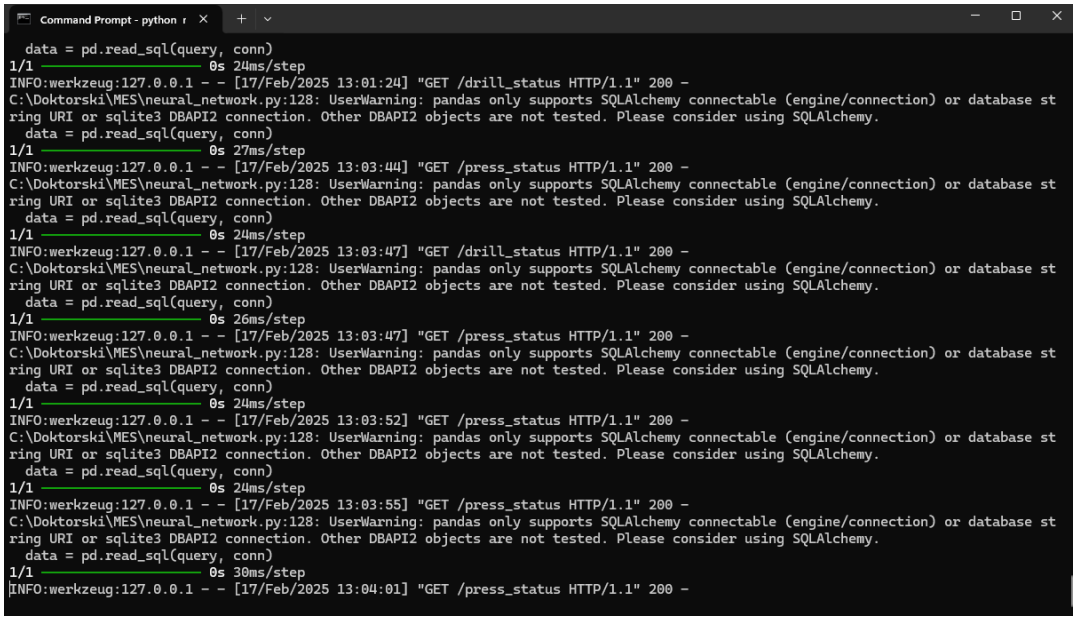
Slika 56 Čitanje stanja u bazi i slanje obavijesti Camundi, nakon što je prva narudžba gotova, a druga čeka potvrdu

Stanje procesa se prati i u konzoli od Camunde, a to je vidljivo na slici 57 u nastavku koja prikazuje izvođenje procesa u Camundi nakon što je prva narudžba gotova, a druga čeka potvrdu završetka ljudske aktivnosti u Camundinoj platformi.



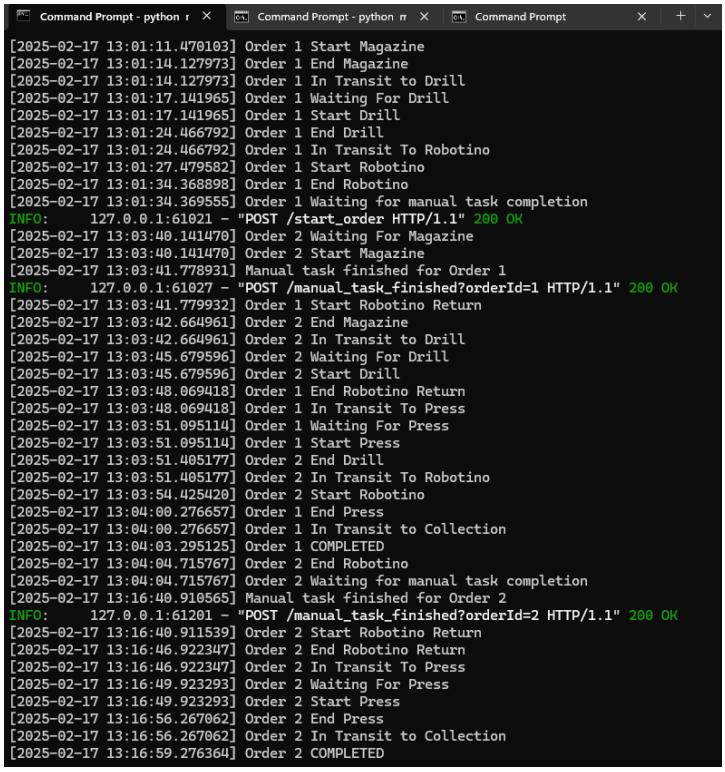
Slika 57 Izvođenje procesa u Camundi nakon što je prva narudžba gotova, a druga čeka potvrdu

Na slici 58 u nastavku se vidi izvođenje neuronske mreže tijekom odvijanja proizvodnje druge narudžbe, odnosno, prikazuju se rezultati treninga neuronske mreže koji su označeni kao 1/1 [0s 24ms/step] – što pokazuje da se izvršava jedan korak po epohi, koji traje nekoliko milisekundi (npr., 24ms, 27ms, 30ms).



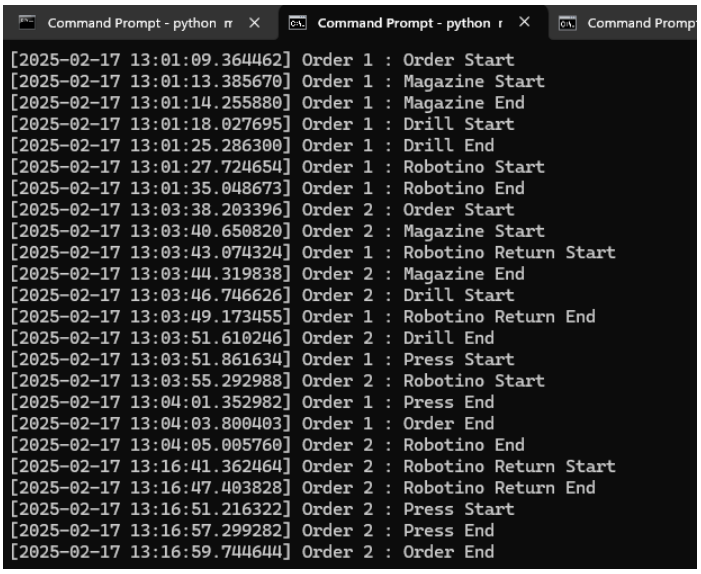
Slika 58 Prikaz neuronske mreže prije nego se završi druga narudžba

Završetak druge narudžbe je vidljiv na slici 59 u nastavku koja prikazuje stanje terminala kako je tekao proizvodni proces. Na slici 59 možemo vidjeti početak svake narudžbe, faze proizvodnje kroz koje je tekla svaka narudžba te kada je bio kraj svake narudžbe. Ujedno je ovaj prikaz i digitalni rezultat da je svaka narudžba uspješno završila. Možda je posebno za istaknuti kako se u terminalu vidi i izvođenje ljudske aktivnosti, koja ne traje uvijek jednako jer je to aktivnost koja je podložna ljudskoj intervenciji i nije automatizirana već je sklona varijacijama.



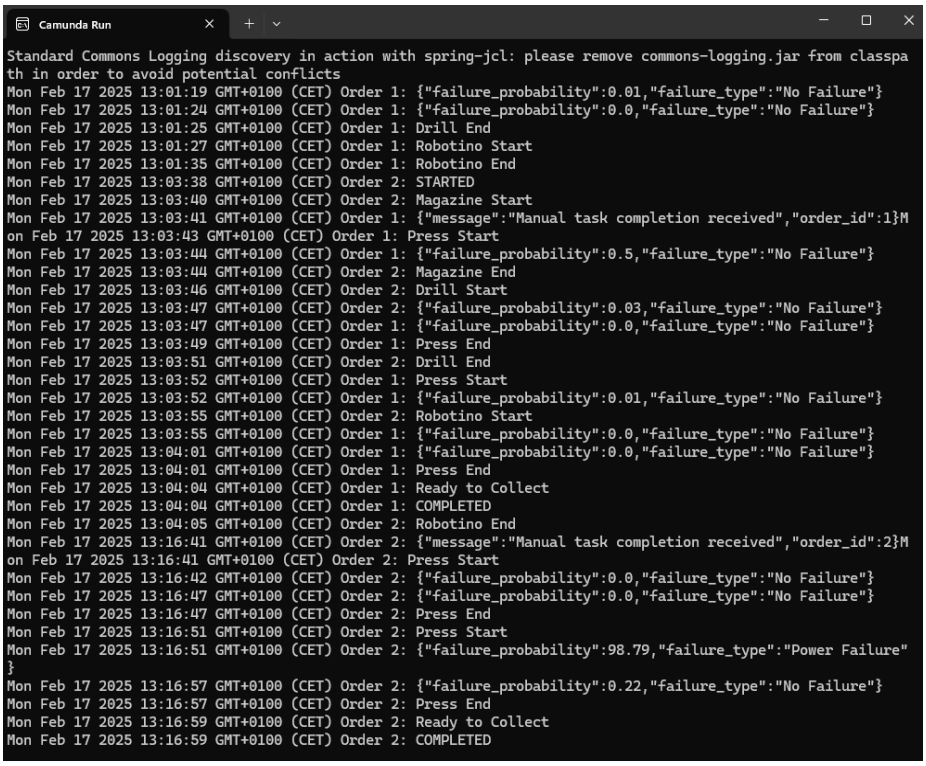
Slika 59 Prikaz završetka procesa proizvodnje nakon druge narudžbe

Nakon završetka proizvodnje obje narudžbe, moguće je pratiti i stanje u bazi. Sljedeća slika 60 prikazuje stanje baze u terminalu gdje se točno vidi veza *vrijeme – narudžba – resurs*, odnosno, vidi se koji je resurs korišten za proizvodnju koje narudžbe u kojem trenutku.



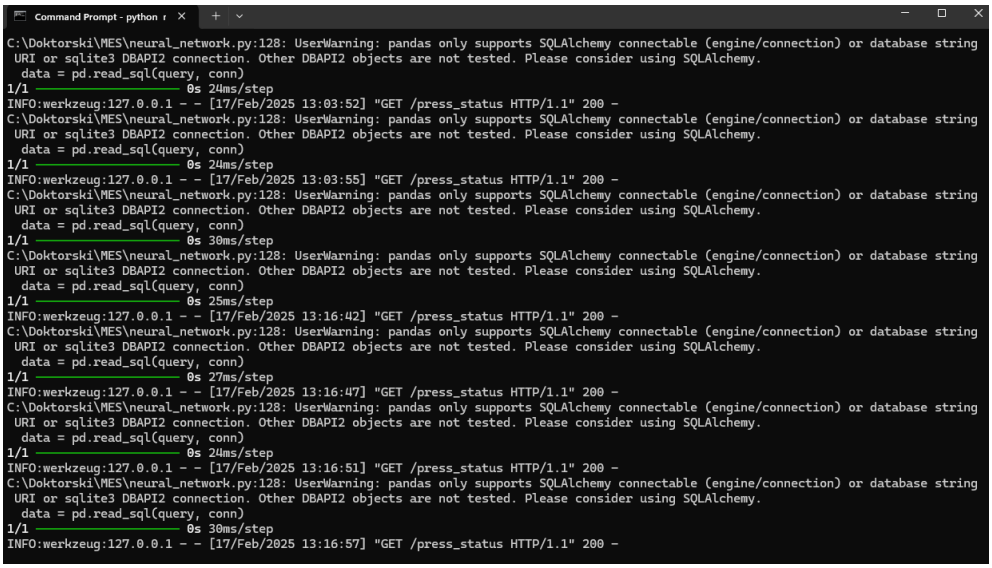
Slika 60 Stanje u bazi nakon završetka druge narudžbe

Slika 61 u nastavku prikazuje log zapis od Camunde nakon završetka obje narudžbe zajedno s podacima o prediktivnom održavanju. U logu su vidljivi podaci za narudžbu 1 i narudžbu 2, pri čemu svaka narudžba prolazi kroz sve aktivnosti definirane modelom digitalnog blizanca te završava s oznakom *COMPLETED*. Tijekom izvođenja aktivnosti, bilježe se informacije o vjerojatnosti kvara (eng., *failure_probability*) i vrsti kvara (eng., *failure_type*) za aktivnosti koje uključuju bušilicu i prešu. Za narudžbu 1, sve faze procesa su prošle bez problema, s niskom vjerojatnošću kvara i oznakom „*No Failure*“. Narudžba je uspješno dovršena u 13:04:01. Kod narudžbe 2 također je većina aktivnosti prošla bez problema, no kod aktivnosti *Press Start* zabilježena je visoka vjerojatnost kvara od 98,79%, s tipom kvara „*Power Failure*“. Unatoč tome, narudžba je također uspješno završena u 13:16:59, ali to znači da preša zahtijeva održavanje u bliskoj budućnosti ili barem intervenciju radnika da provjeri stanje preše. U logu se vide i poruke o ljudskoj aktivnosti, odnosno, ručnom dovršavanju aktivnosti. U konačnici, slika prikazuje uspješan završetak dviju narudžbi te informaciju o potrebnom održavanju preše u budućnosti čime se stavlja naglasak na važnost digitalnog blizanca s prediktivnim održavanjem koji upozorava na moguće probleme u proizvodnji.



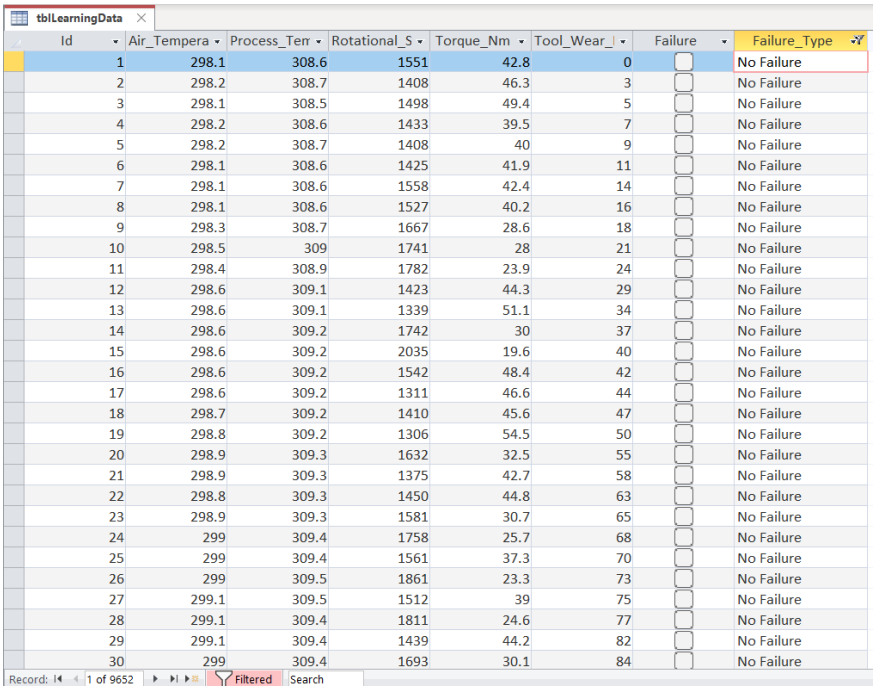
Slika 61 Rezultat izvođenja proizvodnog digitalnog blizanca u Camundi nakon završetka obje narudžbe

Slika 62 u nastavku prikazuje stabilan rad neuronske mreže u stvarnom vremenu nakon završetka proizvodnje dviju narudžbi.



Slika 62 Prikaz neuronske mreže nakon što su završene obje narudžbe

Sljedeća slika 63 prikazuje podatke iz Access baze za učenje neuronske mreže s filterom na prikaz podataka pod uvjetima kada se ne očekuje ispad, odnosno kvar. Ukupno takvih zapisa u bazi ima otprilike 9.650 od početnih 10.000.



tblLearningData

Id	Air_Tempera	Process_Tem	Rotational_S	Torque_Nm	Tool_Wear_	Failure	Failure_Type
1	298.1	308.6	1551	42.8	0		No Failure
2	298.2	308.7	1408	46.3	3		No Failure
3	298.1	308.5	1498	49.4	5		No Failure
4	298.2	308.6	1433	39.5	7		No Failure
5	298.2	308.7	1408	40	9		No Failure
6	298.1	308.6	1425	41.9	11		No Failure
7	298.1	308.6	1558	42.4	14		No Failure
8	298.1	308.6	1527	40.2	16		No Failure
9	298.3	308.7	1667	28.6	18		No Failure
10	298.5	309	1741	28	21		No Failure
11	298.4	308.9	1782	23.9	24		No Failure
12	298.6	309.1	1423	44.3	29		No Failure
13	298.6	309.1	1339	51.1	34		No Failure
14	298.6	309.2	1742	30	37		No Failure
15	298.6	309.2	2035	19.6	40		No Failure
16	298.6	309.2	1542	48.4	42		No Failure
17	298.6	309.2	1311	46.6	44		No Failure
18	298.7	309.2	1410	45.6	47		No Failure
19	298.8	309.2	1306	54.5	50		No Failure
20	298.9	309.3	1632	32.5	55		No Failure
21	298.9	309.3	1375	42.7	58		No Failure
22	298.8	309.3	1450	44.8	63		No Failure
23	298.9	309.3	1581	30.7	65		No Failure
24	299	309.4	1758	25.7	68		No Failure
25	299	309.4	1561	37.3	70		No Failure
26	299	309.5	1861	23.3	73		No Failure
27	299.1	309.5	1512	39	75		No Failure
28	299.1	309.4	1811	24.6	77		No Failure
29	299.1	309.4	1439	44.2	82		No Failure
30	299	309.4	1693	30.1	84		No Failure

Record: 1 of 9652 Filtered Search

Slika 63 Prikaz podataka za treniranje neuronskih mreža kada nema kvara (No Failure)

Slika 64 u nastavku prikazuje podatke iz Access baze za učenje neuronske mreže s filterom na prikaz podataka kada se očekuje kvar uslijed napajanja. Ukupno takvih zapisa u bazi ima 95 od početnih 10.000.

tblLearningData

Id	Air_Tempera	Process_Tem	Rotational_S	Torque_Nm	Tool_Wear_	Failure	Failure_Type
51	298.9	309.1	2861	4.6	143	☑	Power Failure
70	298.9	309	1410	65.7	191	☑	Power Failure
169	298.4	308.3	1433	62.3	20	☑	Power Failure
195	298.2	308.5	2678	10.7	86	☑	Power Failure
208	298.4	308.7	1421	60.7	119	☑	Power Failure
260	298.1	308.2	1420	63.9	19	☑	Power Failure
381	297.5	308.3	2564	12.8	127	☑	Power Failure
443	297.4	308.5	1399	61.5	61	☑	Power Failure
464	297.4	308.7	2874	4.2	118	☑	Power Failure
604	297.9	309.8	1336	71.6	31	☑	Power Failure
848	296.4	307.4	2833	5.6	213	☑	Power Failure
881	295.8	306.3	1235	76.2	89	☑	Power Failure
904	295.7	306.2	2270	14.6	149	☑	Power Failure
1017	296.3	307.2	1319	68.3	24	☑	Power Failure
1096	296.9	307.5	2721	9.3	18	☑	Power Failure
1124	296.6	307.7	1386	62.3	100	☑	Power Failure
1125	296.7	307.8	1258	69	105	☑	Power Failure
1145	297	307.9	1296	69.1	153	☑	Power Failure
1285	298.4	309.5	2153	15.3	98	☑	Power Failure
1325	298.8	310.1	1243	74.5	194	☑	Power Failure
1392	298.9	310.2	2737	8.8	142	☑	Power Failure
1493	298	308.7	1479	58.5	176	☑	Power Failure
1497	298	308.7	1268	69.4	189	☑	Power Failure
1785	298.3	308	2886	3.8	57	☑	Power Failure
1790	298.2	307.9	1313	67.3	67	☑	Power Failure
1809	298.1	307.7	2567	12.8	125	☑	Power Failure
2016	298.5	308.2	1301	66.3	42	☑	Power Failure
2126	299.3	308.9	1258	69.4	119	☑	Power Failure
2234	299.5	308.7	2549	13	179	☑	Power Failure
2300	299.4	308.7	1446	62.9	134	☑	Power Failure

Record: 1 of 95 Filtered Search

Slika 64 Prikaz podataka za treniranje neuronskih mreža s kvarom uslijed napajanja (eng., Power Failure)

Slika 65 u nastavku prikazuje podatke iz Access baze za učenje neuronske mreže s filterom na prikaz podataka kada se očekuje kvar u vidu trošenja alata (eng., *Tool Wear Failure*). Ukupno takvih zapisa u bazi ima 45 od početnih 10.000.

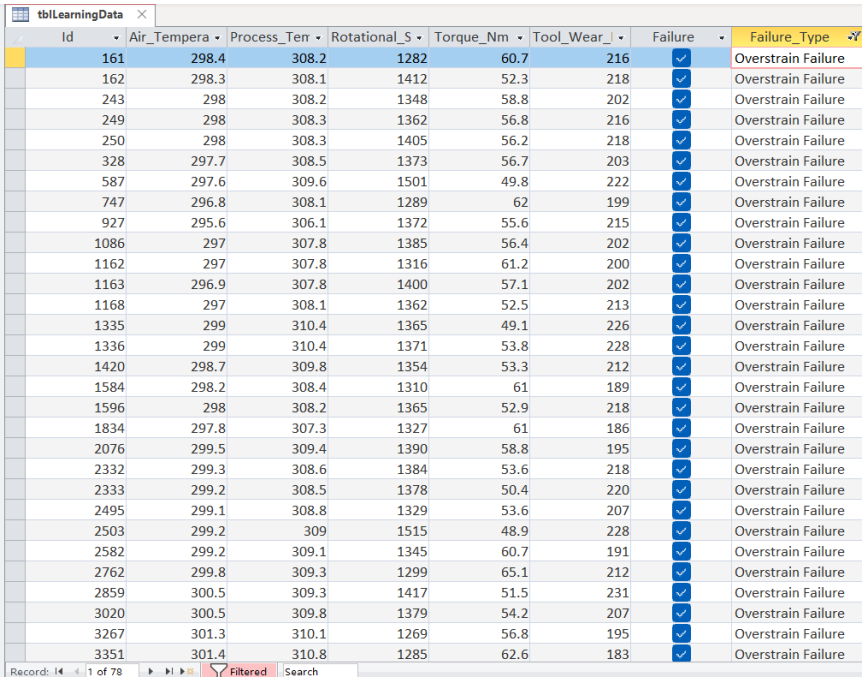
tblLearningData

Id	Air_Tempera	Process_Tem	Rotational_S	Torque_Nm	Tool_Wear_	Failure	Failure_Type
78	298.8	308.9	1455	41.3	208	☑	Tool Wear Failure
1088	296.9	307.8	1549	35.8	206	☑	Tool Wear Failure
1510	298	308.5	1429	37.7	220	☑	Tool Wear Failure
1683	297.9	307.4	1604	36.1	225	☑	Tool Wear Failure
1764	298.2	307.6	1511	31	209	☑	Tool Wear Failure
1997	298.4	308	1416	38.2	198	☑	Tool Wear Failure
2167	299.6	309.2	1867	23.4	225	☑	Tool Wear Failure
2245	299.3	308.4	1542	37.5	203	☑	Tool Wear Failure
2672	299.7	309.3	1399	41.9	221	☑	Tool Wear Failure
2865	300.6	309.4	1380	47.6	246	☑	Tool Wear Failure
2942	300.7	309.6	1996	19.8	203	☑	Tool Wear Failure
3530	301.9	310.9	1567	39	214	☑	Tool Wear Failure
3612	301.7	310.9	1405	46.4	207	☑	Tool Wear Failure
3696	302.2	311.3	1530	37.3	207	☑	Tool Wear Failure
3866	302.6	311.5	1629	34.4	228	☑	Tool Wear Failure
4035	302	310.8	1615	29	235	☑	Tool Wear Failure
4208	302.3	310.9	1710	30.4	218	☑	Tool Wear Failure
4386	301.8	309.7	1442	45.3	233	☑	Tool Wear Failure
4470	302.7	310.5	1446	40.9	211	☑	Tool Wear Failure
4647	303.3	311.4	1497	46	234	☑	Tool Wear Failure
4817	303.4	312	1521	35.9	215	☑	Tool Wear Failure
5142	304.4	313.7	1509	35	205	☑	Tool Wear Failure
5310	303.9	313.2	1422	48	215	☑	Tool Wear Failure
5402	302.6	312.3	1454	54.8	253	☑	Tool Wear Failure
6000	300.4	310.2	1671	30.5	234	☑	Tool Wear Failure
6173	300.8	310.6	1577	37.9	227	☑	Tool Wear Failure
6257	301	310.6	1493	37.8	206	☑	Tool Wear Failure
6341	300.5	309.9	1397	45.9	210	☑	Tool Wear Failure
6420	300.3	309.9	1394	46.7	210	☑	Tool Wear Failure
6760	301.7	311	1441	44.3	208	☑	Tool Wear Failure

Record: 1 of 45 Filtered Search

Slika 65 Prikaz podataka za treniranje neuronskih mreža s kvarom uslijed trošenja alata (eng., Tool Wear Failure)

Slika 66 u nastavku prikazuje podatke iz Access baze za učenje neuronske mreže s filterom na prikaz podataka kada se očekuje kvar uslijed preopterećenja (eng., *Overstrain Failure*). Ukupno takvih zapisa u bazi ima 78 od početnih 10.000.



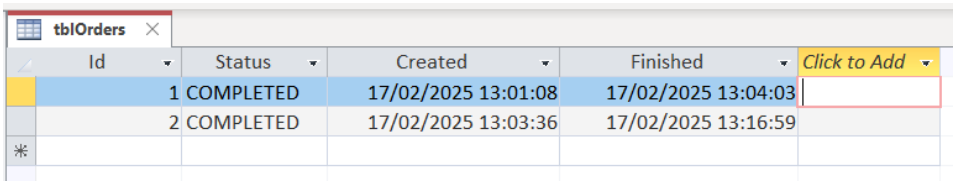
tblLearningData

Id	Air_Tempera	Process_Tem	Rotational_S	Torque_Nm	Tool_Wear_	Failure	Failure_Type
161	298.4	308.2	1282	60.7	216	☑	Overstrain Failure
162	298.3	308.1	1412	52.3	218	☑	Overstrain Failure
243	298	308.2	1348	58.8	202	☑	Overstrain Failure
249	298	308.3	1362	56.8	216	☑	Overstrain Failure
250	298	308.3	1405	56.2	218	☑	Overstrain Failure
328	297.7	308.5	1373	56.7	203	☑	Overstrain Failure
587	297.6	309.6	1501	49.8	222	☑	Overstrain Failure
747	296.8	308.1	1289	62	199	☑	Overstrain Failure
927	295.6	306.1	1372	55.6	215	☑	Overstrain Failure
1086	297	307.8	1385	56.4	202	☑	Overstrain Failure
1162	297	307.8	1316	61.2	200	☑	Overstrain Failure
1163	296.9	307.8	1400	57.1	202	☑	Overstrain Failure
1168	297	308.1	1362	52.5	213	☑	Overstrain Failure
1335	299	310.4	1365	49.1	226	☑	Overstrain Failure
1336	299	310.4	1371	53.8	228	☑	Overstrain Failure
1420	298.7	309.8	1354	53.3	212	☑	Overstrain Failure
1584	298.2	308.4	1310	61	189	☑	Overstrain Failure
1596	298	308.2	1365	52.9	218	☑	Overstrain Failure
1834	297.8	307.3	1327	61	186	☑	Overstrain Failure
2076	299.5	309.4	1390	58.8	195	☑	Overstrain Failure
2332	299.3	308.6	1384	53.6	218	☑	Overstrain Failure
2333	299.2	308.5	1378	50.4	220	☑	Overstrain Failure
2495	299.1	308.8	1329	53.6	207	☑	Overstrain Failure
2503	299.2	309	1515	48.9	228	☑	Overstrain Failure
2582	299.2	309.1	1345	60.7	191	☑	Overstrain Failure
2762	299.8	309.3	1299	65.1	212	☑	Overstrain Failure
2859	300.5	309.3	1417	51.5	231	☑	Overstrain Failure
3020	300.5	309.8	1379	54.2	207	☑	Overstrain Failure
3267	301.3	310.1	1269	56.8	195	☑	Overstrain Failure
3351	301.4	310.8	1285	62.6	183	☑	Overstrain Failure

Record: 1 of 78 Filtered Search

Slika 66 Prikaz podataka za treniranje neuronskih mreža s kvarom uslijed preopterećenja (eng., Overstrain Failure)

Na slici 67 u nastavku je prikazan broj pokrenutih narudžbi u proizvodnji, njihov status (da li su završene ili nisu), vrijeme početka narudžbe i vrijeme završetka narudžbe. Prva narudžba je trajala 2 minute i 55 sekundi, a druga narudžba je trajala 13 minuta i 23 sekunde. Razlika u trajanju je radi ljudske aktivnosti u procesu, točnije, aktivnosti doktorandice koja je tijekom izvođenja druge narudžbe radila slike ekrana aplikacija kako bi dokumentirala tijek proizvodnog procesa za potrebe ovog doktorata. Pritom je bilo potrebno napraviti slike ekrana sučelja Camunde u dijelu ljudske aktivnosti, napraviti sliku ekrana od MES-a, potvrditi završetak aktivnosti na formi u Camundi, poslati poruku MES sustavu o završetku ljudske aktivnosti (gumb na Robotinu) te dodatno spremiti slike ekrana za evidenciju. Sve te radnje dovele su do zadržavanja Robotina, koji je morao čekati završetak ljudske intervencije prije nego što je mogao nastaviti s procesom, što je uzrokovalo produljenje ukupnog trajanja druge narudžbe. Iako se u ovom slučaju produženje dogodilo zbog specifičnih zahtjeva istraživanja, takve situacije su moguće i u stvarnom proizvodnom procesu, primjerice, kada radnik treba obaviti dodatne administrativne radnje, rješavati nepredviđene situacije ili se zadržava u interakciji sa sustavom. Stoga, razlike u trajanju uzrokovane ljudskom komponentom predstavljaju realnu i očekivanu pojavu u proizvodnim poslovnim procesima, što digitalni blizanac upravo omogućuje prepoznati i analizirati. U slučaju potrebe, u ovom dijelu bi se optimizirao rad proizvodnog digitalnog blizanca što je aktivnost 7 iz predložene metodike. Međutim, nije uočena potreba za podešavanjem modela digitalnog blizanca u alatu za izradu procesno servisnih arhitektura.

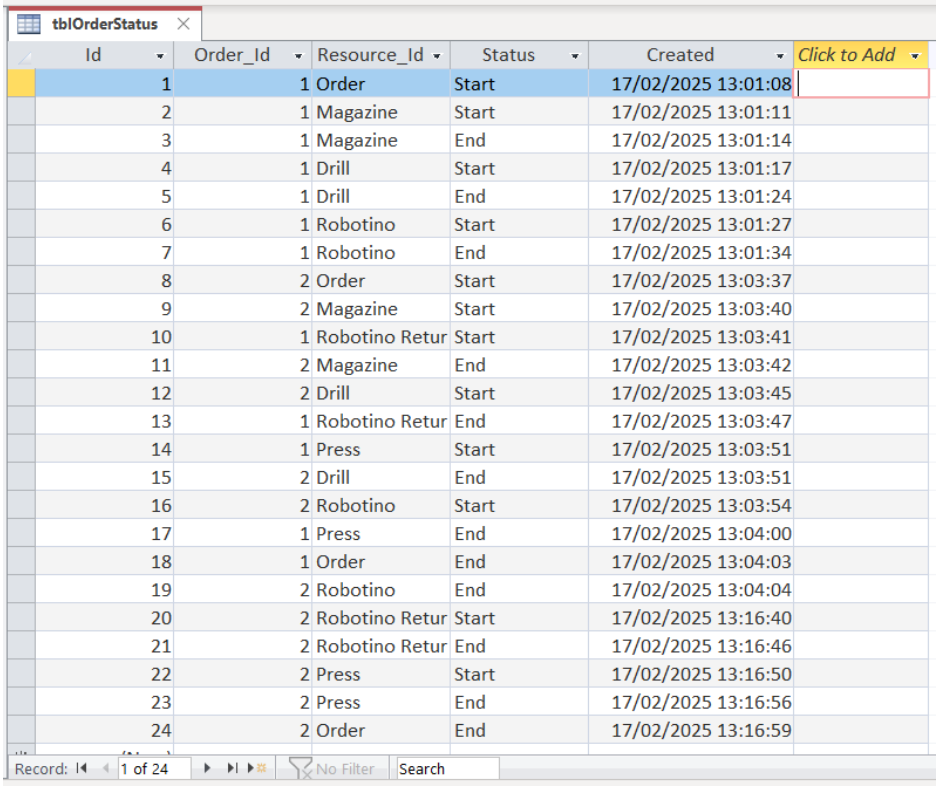


tblOrders

Id	Status	Created	Finished	Click to Add
1	COMPLETED	17/02/2025 13:01:08	17/02/2025 13:04:03	
2	COMPLETED	17/02/2025 13:03:36	17/02/2025 13:16:59	
*				

Slika 67 Redoslijed i broj narudžbi s početkom i završetkom

Redoslijed korištenja resursa prema tome kako su narudžbe prolazile kroz proizvodni proces se vidi na slici 68 u nastavku. Znači, kako su se proizvodi proizvodili i narudžbe su prolazile kroz proizvodni proces, tako se odvijalo i korištenje resursa, odnosno strojeva. Tablica sa slike ima pet bitnih stupaca: *Id, Order_Id, Resource_Id, Status, Created*. Stupac *Id* sadrži redne brojeve zapisa. Stupac *Order_Id* označava kojoj narudžbi zapis pripada (1 ili 2). *Resource_Id* označava resurs, odnosno, stroj koji se koristi u proizvodnom procesu. Status označava početak (*Start*) ili završetak (*End*) aktivnosti određenog resursa. Stupac *Created* bilježi točan datum i vrijeme događaja. Tablica pokazuje da je proizvodni proces za prvu narudžbu (*Order_Id* = 1) započeo 17.02.2025. u 13:01:08, a druga narudžba (*Order_Id* = 2) započela je u 13:03:37. Vrijeme početka i završetka pojedinih aktivnosti jasno ukazuje na redoslijed aktivnosti u proizvodnom procesu. Druga narudžba završena je u 13:16:59, što je evidentno iz zapisa *Order End*.



tblOrderStatus

Id	Order_Id	Resource_Id	Status	Created	Click to Add
1	1	Order	Start	17/02/2025 13:01:08	
2	1	Magazine	Start	17/02/2025 13:01:11	
3	1	Magazine	End	17/02/2025 13:01:14	
4	1	Drill	Start	17/02/2025 13:01:17	
5	1	Drill	End	17/02/2025 13:01:24	
6	1	Robotino	Start	17/02/2025 13:01:27	
7	1	Robotino	End	17/02/2025 13:01:34	
8	2	Order	Start	17/02/2025 13:03:37	
9	2	Magazine	Start	17/02/2025 13:03:40	
10	1	Robotino Retur	Start	17/02/2025 13:03:41	
11	2	Magazine	End	17/02/2025 13:03:42	
12	2	Drill	Start	17/02/2025 13:03:45	
13	1	Robotino Retur	End	17/02/2025 13:03:47	
14	1	Press	Start	17/02/2025 13:03:51	
15	2	Drill	End	17/02/2025 13:03:51	
16	2	Robotino	Start	17/02/2025 13:03:54	
17	1	Press	End	17/02/2025 13:04:00	
18	1	Order	End	17/02/2025 13:04:03	
19	2	Robotino	End	17/02/2025 13:04:04	
20	2	Robotino Retur	Start	17/02/2025 13:16:40	
21	2	Robotino Retur	End	17/02/2025 13:16:46	
22	2	Press	Start	17/02/2025 13:16:50	
23	2	Press	End	17/02/2025 13:16:56	
24	2	Order	End	17/02/2025 13:16:59	

Record: 1 of 24 No Filter Search

Slika 68 Redoslijed korištenja resursa sukladno protoku narudžbi

15 Usporedba simuliranih modela poslovnih procesa s digitalnim blizancima poslovnih procesa

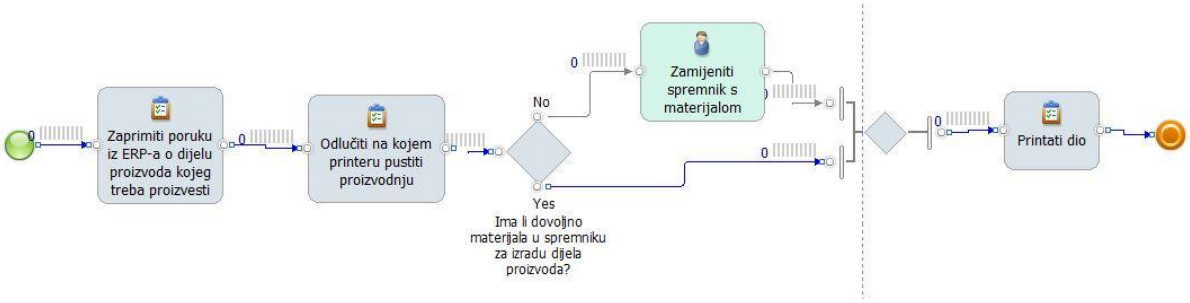
U ovom poglavlju prikazani su rezultati razvijenih simuliranih modela proizvodnih poslovnih procesa za proizvodnju dronova i mobilnih uređaja. To predstavlja 3. aktivnost metodike za razvoj proizvodnih digitalnih blizanaca, koja se odnosi na izradu simuliranih modela proizvodnih poslovnih procesa. Nadalje, u ovom poglavlju će se usporediti rezultati izvođenja proizvodnih digitalnih blizanaca sa simuliranim modelima proizvodnih poslovnih procesa što predstavlja aktivnost 6.2. iz predložene metodike. Procjena točnosti razvijenog digitalnog blizanca provodi se na temelju njegove sposobnosti da vjerodostojno replicira ponašanje stvarnog proizvodnog procesa, pri čemu je važno da se ponašanje digitalnog blizanca podudara s referentnim podacima iz stvarnog okruženja. U ovom istraživanju, procjena točnosti proizvodnih digitalnih blizanaca se može promatrati kroz parametre kao što su vrijeme izvođenja procesa i utrošak resursa, koji su mjerljivi i omogućuju objektivnu usporedbu stvarnog i simuliranog procesa. Podudarnost ovih parametara bi ukazivala na to da je model digitalnog blizanca realističan te da može poslužiti kao pouzdana osnova za daljnje analize i optimizaciju. Za svaki slučaj korištenja je provedena simulacija temeljem povijesnih podataka. Procesne simulacije omogućuju izvođenje *što-ako* (eng., *what-if*) analiza, odnosno ispitivanje različitih budućih ("*TO BE*") varijanti poslovnog procesa bez potrebe za intervencijama u stvarnom sustavu. Provjera točnosti digitalnog blizanca usporedbom sa stvarnim podacima i rezultatima simulacije djeluje kao indikator kvalitete same simulacije. U kontekstu ovog istraživanja, takva provjera bi trebala pokazati da ne postoji značajno odstupanje između simuliranog modela i proizvodnog digitalnog blizanca u pogledu redoslijeda aktivnosti i ključnih vremenskih parametara, čime bi se potvrdilo da digitalni blizanac vjerno replicira ponašanje stvarnog proizvodnog procesa te može poslužiti kao pouzdana osnova za analizu i optimizaciju. Rezultati prikazani u ovom poglavlju služe i kao temelj za potvrdu istraživačke hipoteze: „*Digitalni blizanac proizvodnog poslovnog procesa razvijen proširenom metodikom modeliranja poslovnih procesa emulira realni proizvodni poslovni proces kako se on odvija u stvarnosti*“. Usporedbom ključnih parametara simulacije i digitalnog blizanca proizvodnog poslovnog procesa, kao što su vrijeme trajanja procesa i potrošnja resursa, procjenjuje se razina usklađenosti modela i potvrđuje sposobnost digitalnog blizanca da emulira ponašanje stvarnog sustava.

Modeli poslovnih procesa za koje se provela klasična simulacija temeljem povijesnih podataka su izrađeni u alatu IBM WebSphere Business Modeler verzije 7.0. WebSphere Business Modeler omogućava modeliranje, simulaciju i analizu poslovnih procesa te izradu izvještaja o poslovnim procesima kako bi pomogao u optimizaciji izvedbe poslovnih procesa (IBM, 2025a). Podaci za simulacije su definirani temeljem stvarnih informacija iz proizvodnje. Tako je za definiranje postavki simulacije uzeto okvirno vrijeme trajanja svake aktivnosti u procesu te je broj (ljudskih) resursa definiran temeljem njihove pojavnosti u stvarnom procesu. U nastavku se nalaze rezultati simulacije proizvodnih poslovnih procesa temeljem kojih su izrađeni digitalni blizanci.

15.1 Simulirani model poslovnog procesa proizvodnje dronova

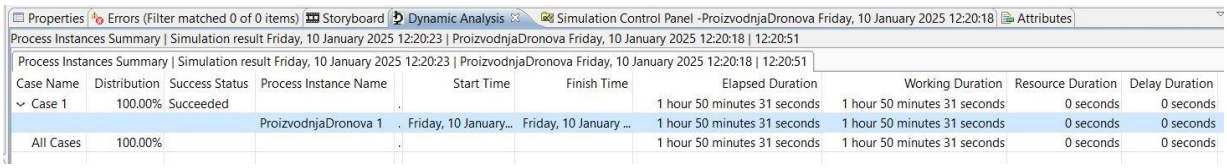
Model poslovnog procesa u nastavku ovog potpoglavlja prikazuje ključne aktivnosti i odluke koje se odvijaju tijekom procesa proizvodnje dronova, od zaprimanja poruke iz ERP-a o dijelu proizvoda kojeg treba proizvesti do printanja dijela proizvoda. Na slici 69 u nastavku je prikazan dijagram poslovnog procesa nakon provedene simulacije, uključujući tijek aktivnosti, točke odluke i njihove povezanosti.

Kako bi se analizirala izvedba procesa, provedena je simulacija s jednom instancom (odnosno, *token*-om). Instanca procesa predstavlja pojedinačni tijek procesa, a njezina putanja istaknuta je na dijagramu plavom bojom. U simulaciju je uključen jedan ljudski resurs (radnik), čija je uloga pokrenuti proizvodnju putem Node-Red sustava te intervenirati u slučaju potrebe, primjerice zamijeniti spremnik s materijalom ako je isti prazan. Simulacija tako omogućava uvid u redoslijed izvršavanja aktivnosti i odluka, te identificiranje mogućih uskih grla ili prilika za optimizaciju.



Slika 69 Simulirani model poslovnog procesa proizvodnje dronova

Slika 70 u nastavku prikazuje rezultate dinamičke analize provedene simulacije poslovnog procesa proizvodnje dronova. Iz prikazanih rezultata vidljivo je da je simulacija uspješno izvedena te su zabilježeni ključni podaci o tijeku procesa. Na slici se može vidjeti vrijeme početka i završetka procesa, ukupno trajanje procesa (eng., *Working Duration*) te informacije o eventualnim kašnjenjima.


Properties | Errors (Filter matched 0 of 0 items) | Storyboard | Dynamic Analysis | Simulation Control Panel -ProizvodnjaDronova Friday, 10 January 2025 12:20:18 | Attributes

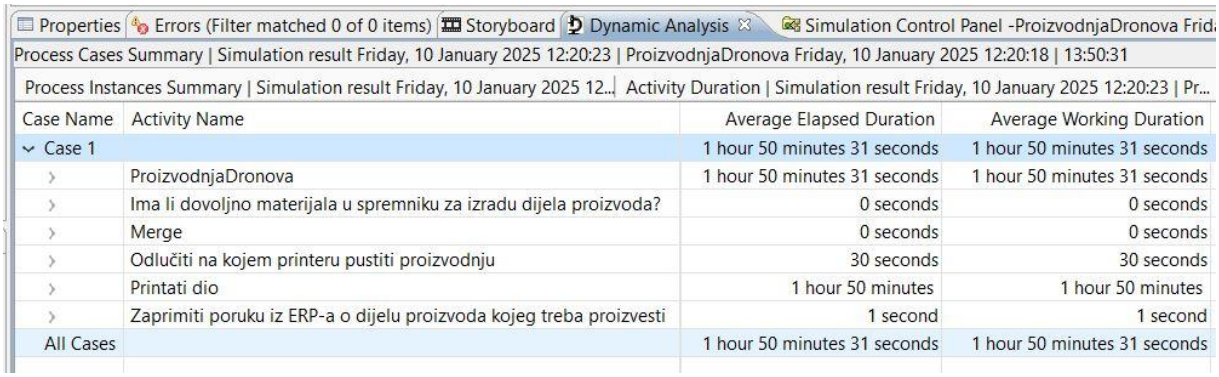
Process Instances Summary | Simulation result Friday, 10 January 2025 12:20:23 | ProizvodnjaDronova Friday, 10 January 2025 12:20:18 | 12:20:51

Process Instances Summary | Simulation result Friday, 10 January 2025 12:20:23 | ProizvodnjaDronova Friday, 10 January 2025 12:20:18 | 12:20:51

Case Name	Distribution	Success Status	Process Instance Name	Start Time	Finish Time	Elapsed Duration	Working Duration	Resource Duration	Delay Duration
Case 1	100.00%	Succeeded				1 hour 50 minutes 31 seconds	1 hour 50 minutes 31 seconds	0 seconds	0 seconds
			ProizvodnjaDronova 1	Friday, 10 January...	Friday, 10 January ...	1 hour 50 minutes 31 seconds	1 hour 50 minutes 31 seconds	0 seconds	0 seconds
All Cases	100.00%					1 hour 50 minutes 31 seconds	1 hour 50 minutes 31 seconds	0 seconds	0 seconds

Slika 70 Rezultati dinamičke analize o uspješnosti provedene simulacije proizvodnje dronova

Nastavak rezultata dinamičke analize simulacije o proizvodnji dronova se mogu vidjeti i na slici 71 u nastavku, koja prikazuje ukupno trajanje svih pokrenutih slučajeva (pokrenut je jedan slučaj jer u stvarnoj proizvodnji printer printa jedan po jedan dio) i trajanje svake aktivnosti koja se izvodila tijekom simulacije, odnosno, prosječno proteklo vrijeme (eng., *Average Elapsed Duration*) te prosječno trajanje rada (eng., *Average Working Duration*). Aktivnost „Printati dio“ traje 1 sat i 50 minuta. U to vrijeme je uključeno zagrijavanje printera koje traje 20 minuta i printanje dijela koje traje 1 sat i 30 minuta. Razlog zašto je ova aktivnost implementirana na ovaj način da nije „djeljiva“ na dvije aktivnosti je taj što se radi o simulaciji koja svakako počiva na povijesnim podacima i izvođenje te aktivnost će trajati svaki puta jednako.



Properties | Errors (Filter matched 0 of 0 items) | Storyboard | Dynamic Analysis | Simulation Control Panel -ProizvodnjaDronova Frid

Process Cases Summary | Simulation result Friday, 10 January 2025 12:20:23 | ProizvodnjaDronova Friday, 10 January 2025 12:20:18 | 13:50:31

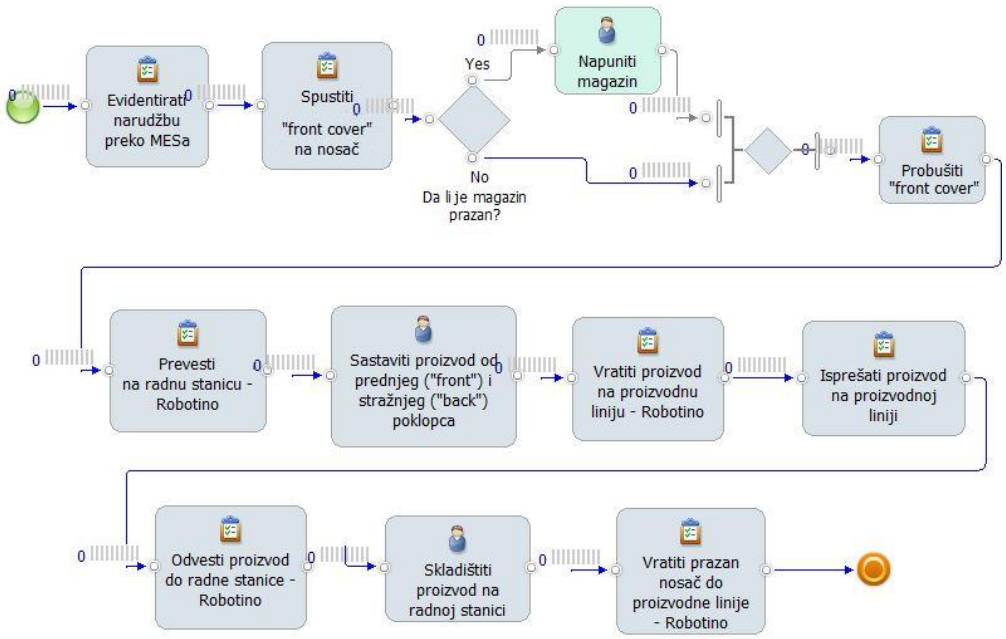
Process Instances Summary | Simulation result Friday, 10 January 2025 12... | Activity Duration | Simulation result Friday, 10 January 2025 12:20:23 | Pr...

Case Name	Activity Name	Average Elapsed Duration	Average Working Duration
Case 1		1 hour 50 minutes 31 seconds	1 hour 50 minutes 31 seconds
	ProizvodnjaDronova	1 hour 50 minutes 31 seconds	1 hour 50 minutes 31 seconds
	Ima li dovoljno materijala u spremniku za izradu dijela proizvoda?	0 seconds	0 seconds
	Merge	0 seconds	0 seconds
	Odlučiti na kojem printeru pustiti proizvodnju	30 seconds	30 seconds
	Printati dio	1 hour 50 minutes	1 hour 50 minutes
	Zaprimiti poruku iz ERP-a o dijelu proizvoda kojeg treba proizvesti	1 second	1 second
All Cases		1 hour 50 minutes 31 seconds	1 hour 50 minutes 31 seconds

Slika 71 Rezultati dinamičke analize s trajanjem aktivnosti procesa proizvodnje dronova

15.2 Simulirani model poslovnog procesa proizvodnje mobilnih uređaja

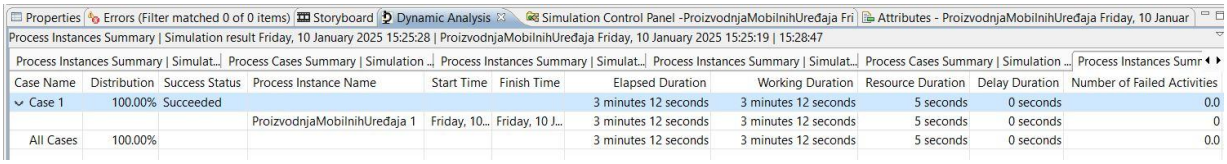
Slika 72 u nastavku prikazuje tijek izvođenja simulacije proizvodnje mobilnih uređaja. Na slici 72 je vidljiva putanja značke, odnosno, instance. Simulacija je provedena korištenjem jedne značke, uz sudjelovanje jednog ljudskog resursa. Njegova uloga je bila punjenje magazina materijalom u slučaju da je magazin prazan, sastavljanje mobilnog uređaja spajanjem prednjeg i stražnjeg poklopca te skladištenje gotovog proizvoda na radnoj stanici nakon prešanja na proizvodnoj liniji. Podaci o trajanju aktivnosti proizvodnog procesa za izradu ovog simuliranog modela su preuzeti iz povijesnih podataka iz MES sustava. Na slici 16 u potpoglavlju *Opis poslovnog procesa proizvodnje mobilnih uređaja* se nalazi prikaz trajanja aktivnosti koje su uključene u izvođenje simulacije u ovom poglavlju.



Slika 72 Simulirani model poslovnog procesa proizvodnje mobilnih uređaja

Na slici 73 u nastavku se nalaze rezultati dinamičke analize napravljene u WebSphere Business Modeleru za simulaciju proizvodnje mobilnih uređaja. Prema rezultatima dinamičke analize, simulacija je završila uspješno te je ukupno trajanje jednog slučaja trajalo 3 minute i 12 sekundi. Na

slici možemo vidjeti vrijeme početka procesa, vrijeme završetka procesa, trajanje rada, utrošak vremena resursa te odgode, koje nije bilo.



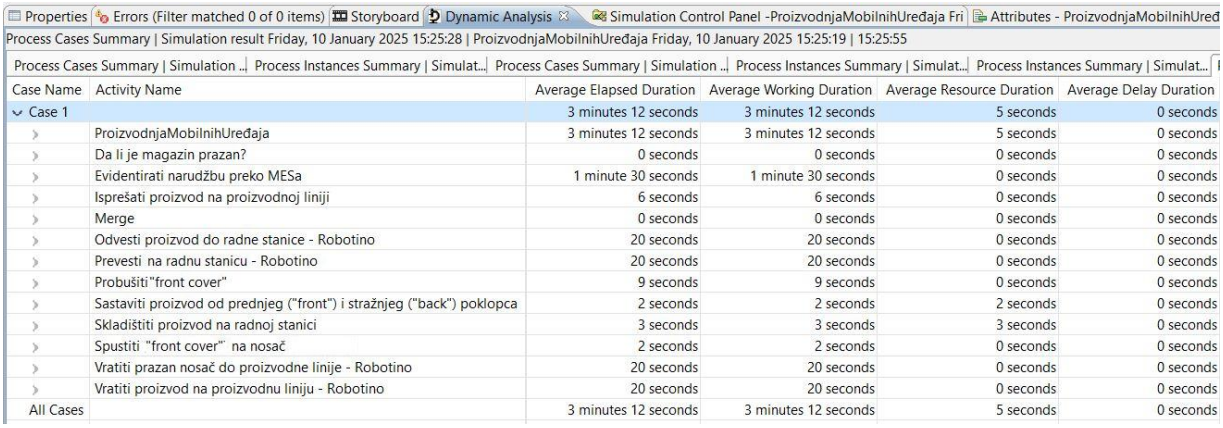
Properties | Errors (Filter matched 0 of 0 items) | Storyboard | Dynamic Analysis | Simulation Control Panel -ProizvodnjaMobilnihUređaja Fri | Attributes - ProizvodnjaMobilnihUređaja Friday, 10 Januar

Process Instances Summary | Simulation result Friday, 10 January 2025 15:25:28 | ProizvodnjaMobilnihUređaja Friday, 10 January 2025 15:25:19 | 15:28:47

Case Name	Distribution	Success Status	Process Instance Name	Start Time	Finish Time	Elapsed Duration	Working Duration	Resource Duration	Delay Duration	Number of Failed Activities
Case 1	100.00%	Succeeded				3 minutes 12 seconds	3 minutes 12 seconds	5 seconds	0 seconds	0.0
			ProizvodnjaMobilnihUređaja 1	Friday, 10...	Friday, 10 J...	3 minutes 12 seconds	3 minutes 12 seconds	5 seconds	0 seconds	0
All Cases	100.00%					3 minutes 12 seconds	3 minutes 12 seconds	5 seconds	0 seconds	0.0

Slika 73 Rezultati dinamičke analize o uspješnosti simulacije proizvodnje mobilnih uređaja

Nastavak rezultata dinamičke analize je prikazan i na slici 74 u nastavku koja prikazuje prosječno trajanje rada za aktivnosti, prosječno trajanje korištenja resursa, prosječno trajanje odgode (ili kašnjenja) za jedan slučaj prolaska simulacije. Pošto je simulacija puštena za jedan slučaj, ukupno trajanje tog jednog slučaja je iznosilo 3 minute i 12 sekundi.
Naime, simulacija nije išla putanjom u kojoj je bilo potrebno napuniti magazin od strane djelatnika jer magazin nije bio prazan. Na skretnici je proces nastavio izvođenje bez punjenja spremnika što je rezultiralo da simulacija kroz aktivnost *Napuniti magazin* nije ni prošla da se evidentira vrijeme trajanja te aktivnosti.



Properties | Errors (Filter matched 0 of 0 items) | Storyboard | Dynamic Analysis | Simulation Control Panel -ProizvodnjaMobilnihUređaja Fri | Attributes - ProizvodnjaMobilnihUređ

Process Cases Summary | Simulation result Friday, 10 January 2025 15:25:28 | ProizvodnjaMobilnihUređaja Friday, 10 January 2025 15:25:19 | 15:25:55

Case Name	Activity Name	Average Elapsed Duration	Average Working Duration	Average Resource Duration	Average Delay Duration
Case 1		3 minutes 12 seconds	3 minutes 12 seconds	5 seconds	0 seconds
	ProizvodnjaMobilnihUređaja	3 minutes 12 seconds	3 minutes 12 seconds	5 seconds	0 seconds
	Da li je magazin prazan?	0 seconds	0 seconds	0 seconds	0 seconds
	Evidentirati narudžbu preko MESa	1 minute 30 seconds	1 minute 30 seconds	0 seconds	0 seconds
	Isprešati proizvod na proizvodnoj liniji	6 seconds	6 seconds	0 seconds	0 seconds
	Merge	0 seconds	0 seconds	0 seconds	0 seconds
	Odvesti proizvod do radne stanice - Robotino	20 seconds	20 seconds	0 seconds	0 seconds
	Prevesti na radnu stanicu - Robotino	20 seconds	20 seconds	0 seconds	0 seconds
	Probušiti "front cover"	9 seconds	9 seconds	0 seconds	0 seconds
	Sastaviti proizvod od prednjeg ("front") i stražnjeg ("back") poklopca	2 seconds	2 seconds	2 seconds	0 seconds
	Skladištiti proizvod na radnoj stanici	3 seconds	3 seconds	3 seconds	0 seconds
	Spustiti "front cover"' na nosač	2 seconds	2 seconds	0 seconds	0 seconds
	Vratiti prazan nosač do proizvodne linije - Robotino	20 seconds	20 seconds	0 seconds	0 seconds
	Vratiti proizvod na proizvodnu liniju - Robotino	20 seconds	20 seconds	0 seconds	0 seconds
All Cases		3 minutes 12 seconds	3 minutes 12 seconds	5 seconds	0 seconds

Slika 74 Rezultati dinamičke analize s trajanjem aktivnosti procesa proizvodnje mobilnih uređaja

Tablična usporedba simuliranih modela poslovnih procesa i proizvodnih digitalnih blizanaca prikazana je u nastavku u tablici 6. Manja razlika u vremenu izvođenja između proizvodnog digitalnog blizanca i simuliranog modela kod proizvodnje mobilnih uređaja proizlazi prvenstveno iz ljudske aktivnosti u procesu, koja u stvarnom proizvodnom okruženju ne traje uvijek jednako dugo, ali i promjenjive brzine kretanja mobilnog robota Robotina, koji povezuje radne stanice u proizvodnom procesu. U simuliranom modelu, izrađenom u IBM WebSphere Business Modeleru, trajanje ljudske aktivnosti i vrijeme kretanja Robotina definirano je kao fiksna vrijednost na temelju povijesnih podataka, pri čemu je korištena prosječna ili okvirna vrijednost trajanja za svaku od tih aktivnosti. Nasuprot tome, u izvođenju proizvodnog digitalnog blizanca u Camundi, trajanje ovih aktivnosti mjeri se u stvarnom vremenu, ovisno o konkretnoj situaciji u trenutku izvođenja procesa. Upravo se u tim segmentima javljaju odstupanja između simuliranog modela i digitalnog blizanca, jer ni ljudska aktivnost ni kretanje mobilnog robota u stvarnosti ne traju uvijek jednako dugo.
Navedeno odstupanje tako dolazi do izražaja u dijelu procesa koji uključuje ručnu montažu proizvoda na radnoj stanici te transport palete s proizvodom između proizvodne linije i radne stanice pomoću Robotina. Radnik u tom dijelu spaja prednji dio maske mobilnog uređaja sa stražnjim dijelom te vraća paletu s proizvodom na Robotina, pri čemu interaktivno sudjeluje s MES sustavom putem sučelja na

Robotinu. Vrijeme trajanja ove aktivnosti u stvarnom procesu može varirati ovisno o spretnosti i brzini radnika, mogućim zastojima pri rukovanju opremom, ili vremenu potrebnom za očitavanje i potvrdu na MES sučelju. Također, vrijeme kretanja Robotina između radnih stanica varira ovisno o uvjetima na putanji, poput pojave prepreka ili smanjenja brzine iz sigurnosnih razloga, što dodatno utječe na ukupno trajanje procesa. Iako je u simuliranom modelu za ovo vrijeme odabran okvirni prosjek, u stvarnom okruženju koje odražava digitalni blizanac, te varijacije su realne i očekivane. Stoga, uočena manja razlika u vremenu između digitalnog blizanca i simuliranog modela nije greška, već rezultat stvarnih oscilacija u ljudskom radu i kretanju autonomnog mobilnog robota.

Broj narudžbe/instance	*Trajanje simuliranog modela*	*Trajanje digitalnog blizanca*
1 – proizvodnja drona	1 sat 50 minuta 31 sekunda	1 sat 50 minuta 14 sekundi
1 – proizvodnja mobitela	3 minute 12 sekundi	2 minute 55 sekundi

Tablica 6 Usporedba simuliranog modela procesa s proizvodnim digitalnim blizancem

Provedeno istraživanje u ovom radu temelji se na simulacijskom eksperimentu kao znanstvenoj metodi, pri čemu su u alatu IBM WebSphere Business Modeler 7.0 izrađeni i simulirani modeli dvaju konkretnih proizvodnih poslovnih procesa: procesa proizvodnje dronova i procesa proizvodnje mobilnih uređaja. Ovi modeli izrađeni su na temelju stvarnih, povijesnih podataka o vremenskom trajanju aktivnosti i dostupnosti resursa. Simulacijski eksperiment proveden je s ciljem analize tijeka i trajanja poslovnih aktivnosti, identifikacije potencijalnih zastoja te evaluacije izvedbe proizvodnih procesa. U simulacijama su puštene instance procesa (po jedan *token*), koje su pokazale odvijanje proizvodnje kroz sve ključne točke proizvodnje. Svaka simulacija rezultirala je dinamičkim podacima s vremenskim tokovima procesa, trajanju pojedinih aktivnosti i opterećenju resursa, što je poslužilo kao osnova za usporedbu s proizvodnim digitalnim blizancima razvijenima u Camundi.

Specifičnost ovog simulacijskog eksperimenta je u tome što su podaci korišteni za simulaciju preuzeti iz realnih proizvodnih sustava, a simulirani modeli u IBM WebSphereu korišteni su kao referentna točka za provjeru vjerodostojnosti razvijenih proizvodnih digitalnih blizanaca u Camundi. Digitalni blizanci, razvijeni za iste proizvodne procese, odražavaju realno vrijeme izvršenja aktivnosti, a podaci o tijeku procesa prikupljeni su putem Python skripti koje dohvaćaju informacije iz relacijske baze podataka. Cilj simulacijskog eksperimenta bio je utvrditi postoji li podudarnost između simuliranih modela i proizvodnih digitalnih blizanaca, pri čemu su rezultati pokazali da između njih nema značajnih odstupanja u smislu redoslijeda provođenja aktivnosti i trajanja aktivnosti, što znači da modeli odražavaju gotovo isti redoslijed aktivnosti, trajanja procesa i opterećenje resursa, čime je potvrđeno da proizvodni digitalni blizanci u Camundi vjerno reproduciraju ponašanje simuliranih modela i time realnog poslovnog procesa.

Rezultati simulacija uspoređeni su s izvođenjem proizvodnih digitalnih blizanaca, pri čemu je utvrđeno da se digitalni blizanci u redoslijedu aktivnosti ponašaju u skladu s realnim poslovnim procesima i simuliranim modelima, dok su uočena manja odstupanja u trajanju pojedinih aktivnosti kod proizvodnje mobilnih uređaja. Ta odstupanja prvenstveno proizlaze iz varijabilnosti ljudske aktivnosti na radnoj stanici, gdje radnik ručno montira proizvod, te promjenjive brzine kretanja mobilnog robota Robotina između radnih stanica. U stvarnom izvođenju procesa, radniku nije uvijek potrebno jednako vrijeme za uzimanje dijelova, montažu i vraćanje palete, kao što ni Robotino ne

prolazi uvijek jednako brzo istu putanju, što znači da njegova brzina može ovisiti o prisutnosti prepreka ili drugim uvjetima u proizvodnom okruženju.

Usporedba simuliranih modela poslovnih procesa s digitalnim blizancima ima višestruku svrhu u okviru istraživanja, metodike i operativne primjene. Na razini istraživanja, ona služi kao ključna verifikacijska aktivnost kojom se procjenjuje točnost i vjerodostojnost digitalnog blizanca, čime se znanstveno potvrđuje njegova sposobnost da realistično emulira ponašanje stvarnog sustava. U kontekstu predložene metodike za razvoj proizvodnih digitalnih blizanaca, usporedba predstavlja integrirani korak validacije modela jer omogućuje povratnu informaciju o kvaliteti izvedenih modela, služeći kao osnova za potencijalno podešavanje simulacijskih parametara. Operativno, ova usporedba ima praktičnu vrijednost u održavanju i nadzoru digitalnog blizanca jer omogućuje stalnu provjeru podudarnosti između stvarnog i simuliranog ponašanja proizvodnog procesa, što je presudno za donošenje operativnih odluka, rano otkrivanje odstupanja i pravovremene intervencije u proizvodnom okruženju. Tako usporedba označava osnovu za kontinuirano poboljšavanje učinkovitosti digitalnog blizanca u stvarnom poslovnom kontekstu. Rezultati provedene usporedbe pokazali su podudarnost između simuliranih modela i proizvodnih digitalnih blizanaca. U slučaju proizvodnje dronova, odstupanje u ukupnom trajanju procesa iznosilo je samo 17 sekundi (1:50:31 u simulaciji naspram 1:50:14 u digitalnom blizancu), dok je kod proizvodnje mobilnih uređaja razlika iznosila 17 sekundi u korist digitalnog blizanca (3:12 u simulaciji naspram 2:55 u stvarnom izvođenju). Osim trajanja, podudarao se i redoslijed izvođenja aktivnosti, čime se potvrđuje dosljednost u logici izvođenja procesa. Također, broj i iskorištenost resursa u oba slučaja pokazali su minimalna odstupanja, što dodatno potvrđuje da razvijeni digitalni blizanci vjerno repliciraju stvarne proizvodne procese. Uočena manja odstupanja mogu se pripisati uobičajenim varijacijama u stvarnom okruženju, kao što su ljudske intervencije ili promjenjiva brzina autonomnog mobilnog robota.

Digitalni blizanac proizvodnje mobilnih uređaja razvijen u Camundi, vjerno prikazuje tijek proizvodnje u realnom vremenu, uključujući pokretanje aktivnosti, prijelaz palete između radnih stanica te intervencije radnika na ručnoj radnoj stanici. Tijekom izvođenja simulacija u IBM WebSphere Business Modeleru i digitalnog blizanca u Camundi, nije uočeno odstupanje u redoslijedu aktivnosti, što potvrđuje da su oba modela konzistentna s realnim poslovnim procesom. Razlike u vremenskom trajanju pojedinih aktivnosti pripisuju se realnim uvjetima proizvodnje, što upućuje na to da proizvodni digitalni blizanac ne samo da odražava stvarno ponašanje proizvodnog procesa, već i pruža precizniji uvid u varijabilnost ljudske aktivnosti i kretanje autonomnog mobilnog robota, što simulirani modeli najčešće pojednostavljuju. Tako proizvodni digitalni blizanac predstavlja pouzdan alat za daljnju analizu i optimizaciju proizvodnih poslovnih procesa, s mogućnošću pravovremenog prepoznavanja odstupanja koja u simulacijama nisu vidljiva.

16 Unaprjeđenje postojećih alata za upravljanje poslovnim procesima

U ovom poglavlju analiziraju se funkcionalnosti koje alati za modeliranje poslovnih procesa trebaju podržavati kako bi bili primjenjivi u razvoju proizvodnih digitalnih blizanaca. Poseban naglasak stavlja se na podršku komunikaciji sa stvarnim procesima, integraciju s prediktivnim modelima temeljenima na umjetnoj inteligenciji te obradu podataka u stvarnom vremenu.

U nastavku će se na primjeru BPMN-a i Camunda platforme prikazati konkretna rješenja i nedostaci, no zaključci i preporuke odnose se na sve alate za modeliranje poslovnih procesa. Trenutno, za razvoj proizvodnog digitalnog blizanca nedostaje element koji omogućuje jasno definiranje načina komunikacije sa stvarnim procesom te lokacije na kojoj se proces nalazi, što bi podrazumijevalo, primjerice, URL adresu. Stoga se u ovom radu predlaže proširenje dijela ulaznih konektora (eng., *Connector inputs*) dodavanjem događaja za URL, kojim bi se specificirala adresa za pristup AI api-ju. Slično tome, dio izlaznih konektora (eng., *Connector outputs*) bi bilo potrebno proširiti dodavanjem atributa, odnosno, događaja *apiResponse*, koji bi služio za obradu povratnih informacija dobivenih API pozivom, odnosno podacima koje vraća RESTful servis.

Većina postojećih alata za modeliranje poslovnih procesa ne uključuje standardizirane elemente koji omogućuju obradu odgovora API poziva u kontekstu integracije s vanjskim sustavima (npr. Node-RED) putem REST protokola. Uvođenje takvih elemenata bilo bi ključno za unapređenje podrške BPMN-a za integracije u distribuiranim sustavima temeljenima na modernim komunikacijskim protokolima poput REST-a.

Kako bi se ostvarila komunikacija proizvodnog digitalnog blizanaca s elementom umjetne inteligencije, korištena je mogućnost servisne aktivnosti (eng., *service task*). Jednako tako, za prijenos informacija o izvođenju proizvodnog poslovnog procesa, od Camunde prema vanjskom svijetu u realnom vremenu, korištena je servisna aktivnost. Razlog zašto se koristila servisna aktivnost je taj što se u BPMN-u servisna aktivnost koristi za predstavljanje automatizirane aktivnosti koju izvršava sustav ili vanjska usluga. Ovaj element ključan je za modeliranje modernih poslovnih procesa jer jasno razlikuje ljudske aktivnosti (eng., *human task*) od aktivnosti koje obavljaju automatizirani sustavi. Servisne aktivnosti se često koriste u modeliranju poslovnih procesa radi toga što eksplicitno označavaju da se korak u procesu automatizira. One su idealne za ilustraciju integracija s vanjskim uslugama, poput REST ili SOAP API-jeva za dohvaćanje ili ažuriranje podataka. Korištenje servisnih aktivnosti jasno pokazuje sudionicima poslovnog procesa koji dijelovi procesa ovise o tehnologiji. Na kraju, servisne aktivnosti omogućavaju interakciju u stvarnom vremenu, poput dohvaćanja trenutnih podataka ili pokretanja tijekova rada na temelju vanjskih okidača, što je važno u dinamičnim poslovnim okruženjima.

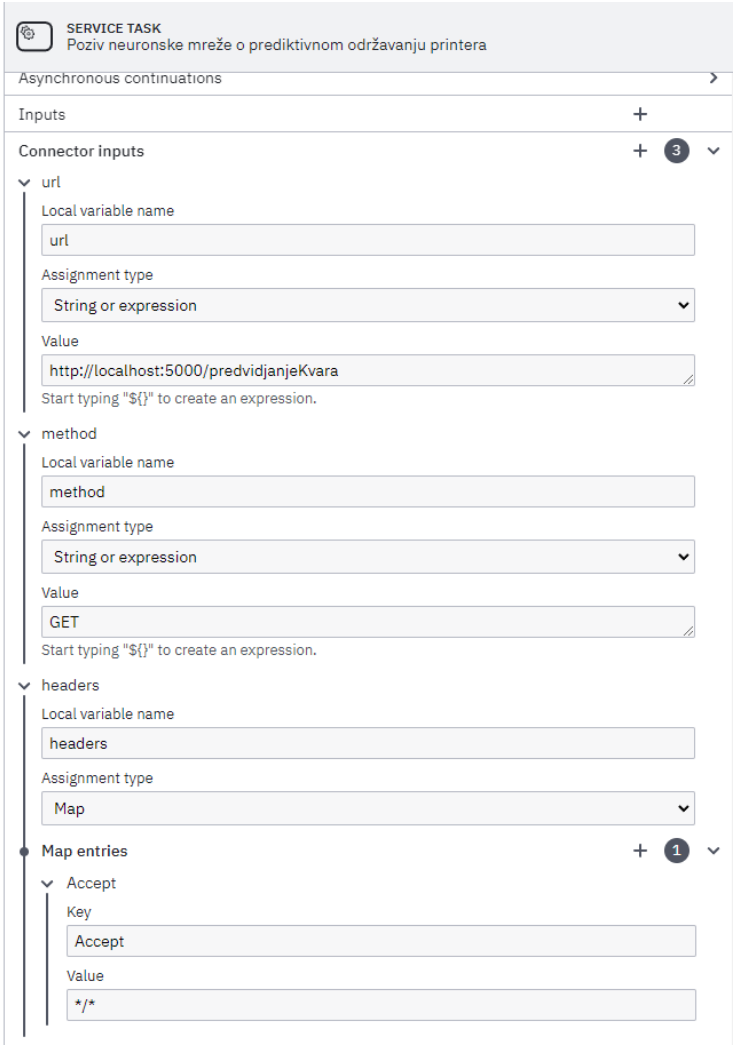
U Camunda Modeleru, implementacija servisne aktivnosti proširena je kako bi podržala preciznu konfiguraciju i integraciju s vanjskim sustavima putem REST API-ja. Ova implementacija uključila je dodavanje specifičnih ulaznih konektora, koji su omogućili fleksibilnost i prilagodbu servisa prema zahtjevima proizvodnog procesa.

Za konfiguraciju servisne aktivnosti korištena su tri ključna ulazna konektora:

- *URL*, koji specificira adresu krajnje točke API-ja koja se poziva čime se osigurava točna lokacija za komunikaciju s vanjskim sustavom.

- *Metoda* (eng., *Method*), koji definira HTTP GET metodu koja se koristi za poziv, omogućujući usklađivanje s funkcionalnostima i protokolima ciljnog API-ja.
- *Zaglavlje* (eng., *Headers*), koji omogućuje dodavanje zaglavlja potrebnih za API poziv, čime se osigurava ispravna autentifikacija, autorizacija i specificiranje formata podataka.

Dodatno, unutar *Headers* konfiguracije dodana je mapa (eng., *Map entries*) s ključem „*Accept*“ i vrijednošću "/". Ova postavka se koristila za signaliziranje API-ju da poziv prihvaća podatke u bilo kojem formatu odgovora, čime se povećava interoperabilnost između sustava.



Slika 75 Konfiguracija servisne aktivnosti: Ulazni konektori

Implementacija servisne aktivnosti proširena je i konfiguracijom izlaznog konektora, čime se omogućilo učinkovito prikupljanje i daljnja obrada podataka dobivenih od vanjskih sustava putem REST API poziva. Ova konfiguracija omogućila je dinamičko upravljanje povratnim vrijednostima, pri čemu se osigurala usklađenost s poslovnim zahtjevima proizvodnog poslovnog procesa i tehničkim specifikacijama procesa. Za potrebe obrade podataka vraćenih od API-ja, unutar servisne aktivnosti konfiguriran je izlazni konektor s nazivom „*apiResponse*“, koji se koristi kao naziv procesne varijable (eng., *process variable name*). Ova varijabla omogućuje spremanje odgovora API-ja za daljnje korištenje u digitalnom blizanacu proizvodnog poslovnog procesa. Tako su postavke izlaznog konektora pod nazivom *apiResponse*:

- *apiResponse,* koji definira varijablu u proizvodnom poslovnom procesu koja omogućuje pristup podacima iz odgovora API-ja.
- tip dodjele (eng., *assignment type*) je „*string or expression*“, koji specificira vrstu vrijednosti koja će biti dodijeljena varijabli, čime se podržava fleksibilnost u radu s različitim tipovima podataka.
- vrijednost je „*${response}*“, pri čemu se ovaj izraz koristi za dohvaćanje podataka iz odgovora API-ja (*response*) i njihovo dodjeljivanje procesnoj varijabli.

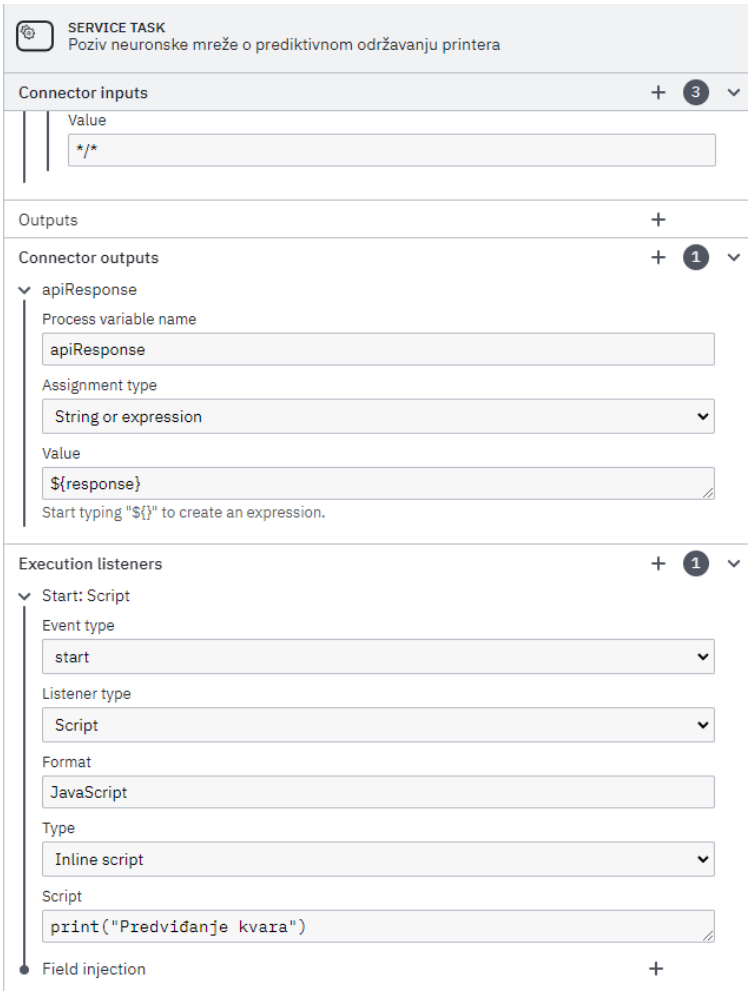
Ova konfiguracija je omogućila proizvodnom digitalnom blizancu pristup ključnim informacijama iz API poziva za daljnju obradu ili donošenje odluka unutar procesa. Znači, dodavanje izlaznog konektora u servisnu aktivnost je podržalo napredan pristup integraciji proizvodnog digitalnog blizanaca s vanjskim sustavom.

Ovakav pristup, kojim se ulazni i izlazni konektori implementiraju na servisnim aktivnostima, omogućuje usklađenost s načelima *stateless* arhitekture. Takva arhitektura osigurava obradu i pohranu podataka na način koji omogućuje neovisno funkcioniranje procesnih komponenti. Da implementacija ulaznih i izlaznih konektora na servisnim aktivnostima podržava ključne principe *stateless* arhitekture, znači da se podaci ne vežu trajno uz određenu procesnu komponentu ili sesiju. Umjesto toga, podaci se obrađuju, prenose i pohranjuju na decentraliziran i neovisan način, čime se osigurava da svaka procesna komponenta može funkcionirati autonomno bez ovisnosti o prethodnom stanju. Ovaj pristup omogućava veću fleksibilnost u dizajnu procesa i olakšava integraciju s vanjskim sustavima jer se podaci razmjenjuju putem definiranih varijabli i konektora. U kontekstu ovog istraživanja to znači da se proces može prilagoditi većim opterećenjima i većom interoperabilnošću, odnosno mogućnošću integracije različitih komponenti unutar digitalnog blizanca proizvodnog poslovnog procesa.

Dodatna funkcionalnost koja je implementirana na servisnim aktivnostima u proizvodnom digitalnom blizancu u Camunda Modeleru je opcija koja osluškuje izvođenje (eng., *Execution listeners*), a koristila se kako bi se omogućilo pokretanje specifične skripte tijekom izvođenja proizvodnog digitalnog blizanaca, čime se dodatno proširila fleksibilnost i prilagodljivost modela digitalnog blizanaca u Camundi. Implementirana skripta unutar opcije za osluškivanje izvođenja pružila je funkcionalnost prikazivanja i obrade rezultata prediktivnog održavanja. Ova skripta je omogućila generiranje obavijesti vezanih uz stanje sustava te je osigurala da ključne informacije o predviđenim kvarovima i potrebnim intervencijama budu dostupne u stvarnom vremenu. Na taj način je model digitalnog blizanca dodatno obogaćen dinamičkim podacima, čime su unaprijeđene njegove mogućnosti analize i podrške u donošenju odluka unutar proizvodnog procesa.

Za implementaciju funkcionalnosti osluškivanja izvođenja, korišten je *Start* događaj koji se aktivirao prilikom početka izvršenja servisne aktivnosti. Konfiguracija funkcionalnosti osluškivanja izvođenja je uključivala nekoliko ključnih parametara:

- *Start* (eng., *Start event*) u kojem je tip događaja (eng., *Event type*) postavljen na "start", što označava da je osluškivač aktiviran kada aktivnost započinje, tj. na početku izvođenja aktivnosti.
- *Tip osluškivača* (eng., *Listener type*) je „*script*“, gdje je odabrano korištenje osluškivača skripti (eng., *script listenera*), što znači da će se izvršiti skripta prilikom pokretanja aktivnosti.
- *Format* je „*java script*“ , što je omogućilo upotrebu JavaScript jezika za implementaciju željene funkcionalnosti unutar Camunda okruženja.
- Skripta (eng., *script*) koja je omogućila ispisivanje poruke u log zapisu.

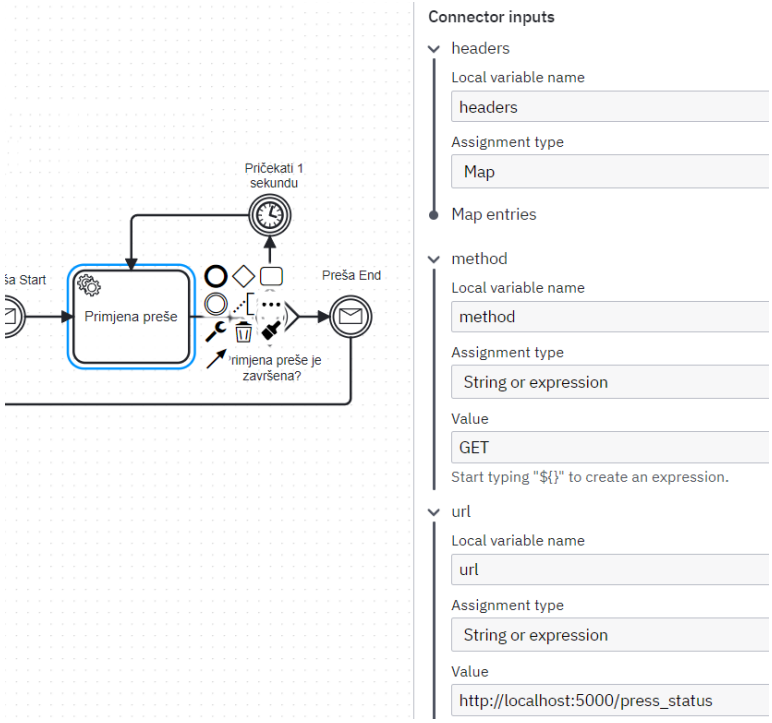


Slika 76 Konfiguracija servisne aktivnosti: Izlazni konektori

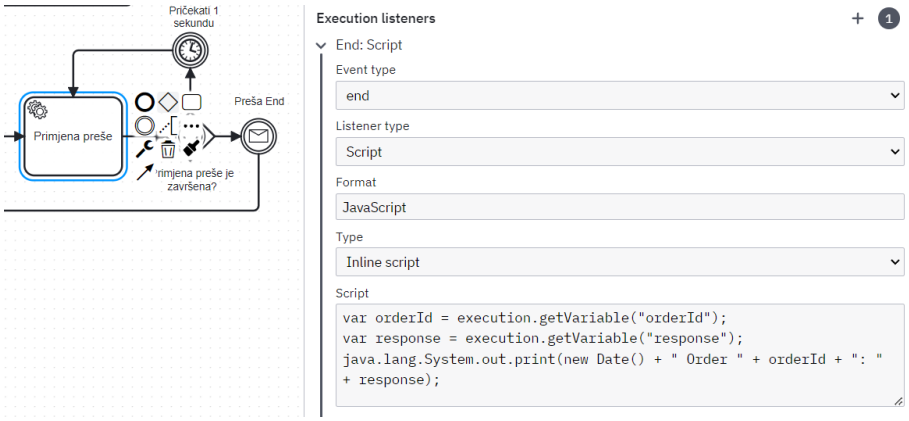
Odabir JavaScript-a kao jezika za skriptu omogućio je jednostavnost implementacije, ali i visoku razinu interoperabilnosti s Camunda okruženjem. JavaScript je široko podržan i nudi bogat skup funkcionalnosti za rad s podacima i interakciju sa sustavima. U kontekstu poslovnih procesa i proizvodnih digitalnih blizanaca, takve skripte se koriste za automatsko bilježenje logova, inicijalizaciju varijabli i za povezivanje s vanjskim sustavima u stvarnom vremenu. Ovaj primjer implementacije u Camundi ilustrira kako bi alat za modeliranje trebao omogućiti jednostavnu integraciju s vanjskim API-jem putem ulaznih i izlaznih konektora. Takva podrška nužna je u svim alatima za modeliranje poslovnih procesa kako bi se omogućila interoperabilnost s vanjskim sustavima u stvarnom vremenu.

S ciljem podrške prediktivnom održavanju u proizvodnim digitalnim blizancima, alati za modeliranje poslovnih procesa trebali bi omogućiti posebnu vrstu aktivnosti koja bi olakšala integraciju prediktivnih modela strojnog učenja u procese. Ova funkcionalnost ne bi smjela biti ograničena na Camundu, već bi trebala biti standardizirani dio svih alata. Prijedlog je uvođenje nove vrste uslužne aktivnosti (eng., *service task*).

Dio s prediktivnim održavanjem preše u slučaju proizvodnje mobilnih uređaja je ovako implementiran u Camundi pomoću niza objekata:



Slika 77 Ulazi konektora na uslužnoj aktivnosti Primjena preše



Slika 78 Slušatelji izvođenja na uslužnoj aktivnosti Primjena preše

Što je ideja nove uslužne aktivnosti i kako bi ona radila je objašnjeno u nastavku. Sustav bi trebao kontinuirano nadzirati stanje procesa pomoću tehnike strojnog učenja sve dok se proizvodni poslovni proces ne završi. Drugim riječima, radilo bi se o „*pollingu*“ , npr., neuronske mreže za stanje procesa. Ovaj nadzor bi bio implementiran kroz uslužnu aktivnost, odnosno, automatizaciju koja bi imala unaprijed definirane parametre. Ti parametri bi se slali AI API-ju, koji bi analizirao podatke. Ako bi sustav otkrio nepravilnost, proces bi se automatski zaustavio, a strojevi i radnici bi odmah dobili obavijest o problemu.

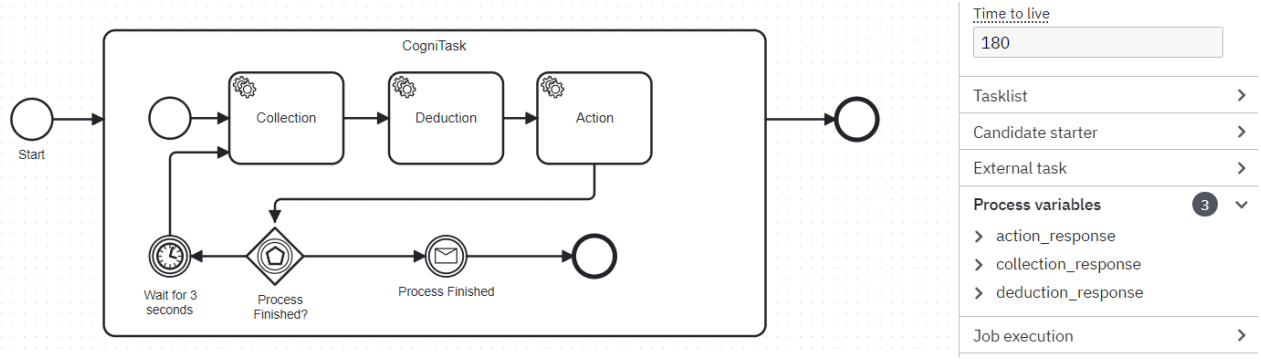
U slučaju aktivnosti poput "Primjena preše", sustav bi pratio stanje preše tijekom rada. Podaci o njenom trenutnom stanju slali bi se, npr., neuronskoj mreži putem API-ja. Sustav umjetne inteligencije bi zatim analizirao podatke i vraćao rezultat, odnosno, procjenu vjerojatnosti kvara i vrstu mogućeg kvara. Na temelju rezultata analize, u istoj uslužnoj aktivnosti mogu se definirati odgovarajuće radnje. Ako sustav umjetne inteligencije procijeni visoku vjerojatnost kvara, može se automatski zaustaviti proces kako bi se spriječila šteta ili se mogu poslati obavijesti nadležnim

osobama koje će poduzeti potrebne mjere. Na taj način sustav omogućuje pravovremenu reakciju i smanjuje rizik od neplaniranih zastoja u proizvodnji.

Ono o čemu treba razmisliti je kako definirati generičke meta parametre (npr., trenutno su ulazi dinamički podaci koji se šalju tijekom odvijanja procesa) nove uslužne aktivnosti pošto je ideja da nova uslužna aktivnost radi automatski, a za to je potreban strukturiran ulaz i strukturiran izlaz. Osim ranije spomenutih ulaznih i izlaznih konektora koji se implementiraju na uslužnim aktivnostima, potreban je i dodatan set pravila kao ulaz koji bi označavao kada proces treba reagirati, a kada ne. Na ovaj način bi se spojila potreba za novim ulaznim konektorom, novim izlaznim konektorom i novom uslužnom aktivnosti u jednu cjelinu, odnosno, novi element koji bi zapravo stvarno bio nova uslužna aktivnost, a koji bi uključivao sve navedeno i sve što je potrebno kako bi alati za modeliranje poslovnih procesa podržali mogućnost prediktivnog održavanja. Da bi se to ostvarilo potrebno je razmisliti o 3 općenita zahtjeva:

1. specifikacija za dohvat relevantnih podataka za sustav umjetne inteligencije (npr., stanje stroja),
2. specifikacija poziva sustava umjetne inteligencije (npr., neuronska mreža),
3. specifikacija za daljnje akcije koja odlučuje i potencijalno reagira temeljem odgovara od sustava umjetne inteligencije.

Prijedlog je da se nova uslužna aktivnost zove CogniTask i da njezina implementacija izgleda kao prikaz na slici 79 u nastavku s razlikom da navedena uslužna aktivnost još ne postoji i kako bi se implementirala njezina funkcionalnost, ona je razvijena u obliku podprocesa koji u sebi implementira sve tri potrebne aktivnosti sa svim proširenjima, ali je preporuka da ovo ne ostane samo na podprocesu, već da se u alate za modeliranje poslovnih procesa uvede novi element, nova uslužna aktivnost, koja bi radila ovo što radi razvijeni CogniTask podproces. Time bi se olakšala implementacija prediktivnog održavanja u proizvodni digitalni blizanac, implementacija bi bila brža i dostupnija svim stručnjacima koji se bave modeliranjem poslovnih procesa bez obzira na prethodno znanje i iskustvo u programiranju.

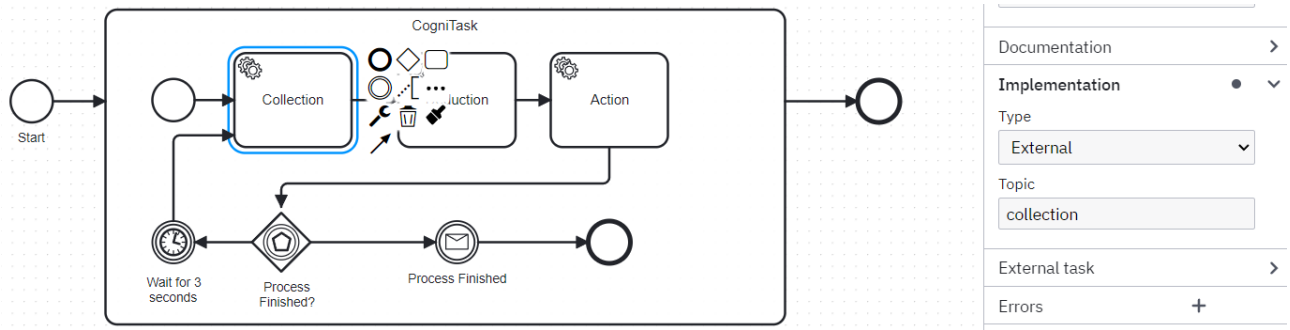


Slika 79 Implementacija CogniTaska u Camundi

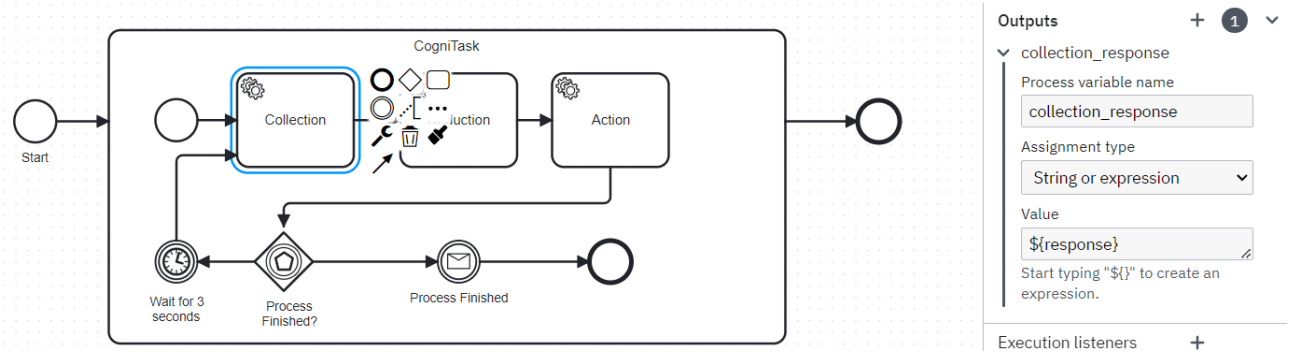
Na prethodnoj slici 79 se nalazi prikaz implementacije CogniTask podprocesa koji se sastoji od tri uslužne aktivnosti koje su vanjskog tipa (eng., *external*) pošto se radi o vanjskim zadacima. Periodički se poziva ciklus tijekom trajanja glavne aktivnosti koja je dio nadogradnje digitalnog blizanca i zbog toga je u podprocesu dodan *timer*. Cilj timera je da se vanjski zadaci pozivaju s odgodom da se ne opterete vanjski sustavi.

Prva aktivnost se naziva „Prikupljanje“ (eng., i na slici *Collection*) i njezina je uloga prikupljanje i obrada sirovih podataka. Funkcija joj je da dohvaća podatke iz API-ja, baza podataka ili senzora te ih priprema za sljedeći korak, a to je dedukcija. Npr., ova aktivnost poziva API za dohvaćanje podataka, provodi prethodnu obradu i sprema podatke kao procesnu varijablu.
Druga aktivnost se naziva „Dedukcija“ (eng., i na slici *Deduction*) i njezina je uloga da komunicira s AI modelom (modelom strojnog učenja) kako bi donijela predikcije. Funkcija joj je da prosljeđuje prikupljene podatke AI servisu, preuzima odgovor i vraća ga procesu. Npr., ova aktivnost šalje zahtjev AI sustavu za analizu ili klasifikaciju i sprema dobiveni odgovor.
Treća, ujedno i zadnja, aktivnost je „Akcija“ (eng., i na slici *Action*). Njezina je uloga što izvršava radnje temeljene na odgovoru AI sustava. Funkcija joj je da analizira rezultate AI zaključivanja i poduzima odgovarajuće korake, poput ažuriranja zapisa, slanja obavijesti ili pozivanja API-ja. Npr., ova aktivnost šalje naredbe nekom sustavu, pokreće alarm ili inicira poslovni proces na temelju AI izlaza.
Znači, npr., zadatak aktivnosti prikupljanja podataka je da dohvaća informacije sa senzora ili drugog izvora i sprema ih u vrijednost *${response}* procesne varijable „*collection_response*“ u Camundi, odnosno, Python skripti. Drugim riječima, u Camundi, varijabla koja se koristi za odgovor naziva se „*collection_response*“. Nakon toga, *${response}* se mapira u vrijednost varijable „*deduction_response*“, kako bi se podaci mogli koristiti u sljedećoj aktivnosti.

Slike 80 i 81 u nastavku prikazuju detalje implementacije aktivnosti „Prikupljanja“ u Camundi. Ekvivalentno bi bilo za „Dedukciju“ i „Akciju“.



Slika 80 Prikaz implementacije za aktivnost Collection



Slika 81 Prikaz implementacije izlaznih odgovora za aktivnost Collection

U Prilogu 9 ovog dokumenta se nalazi Python skripta koja implementira radnika (eng., *worker*) za obradu vanjskih aktivnosti u Camundinoj platformi koristeći *PyCamunda* biblioteku. Skripta omogućuje preuzimanje aktivnosti iz mehanizma vanjske aktivnosti (eng., *external task*), njihovu

obradu kroz definirane funkcije te vraćanje rezultata u model poslovnog procesa. Ova funkcionalnost važna je zbog orkestracije poslovnih procesa, gdje se aktivnosti mogu distribuirati između različitih komponenata sustava, uključujući sustave temeljene na umjetnoj inteligenciji.

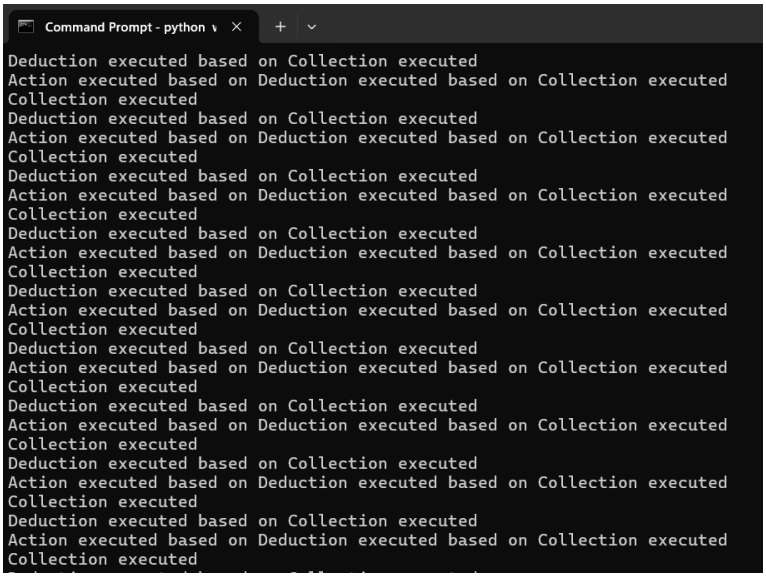
Na početku skripte definira se klasa *ExternalTaskException*, koja omogućuje upravljanje iznimkama u slučaju neuspješne obrade aktivnosti. Ova klasa implementira mehanizam za ponavljanje neuspjelih aktivnosti i određivanje vremena ponovnog pokušaja radi otpornosti sustava.

Glavna klasa nazvana *Worker* predstavlja instancu radnika koji se povezuje s Camunda procesnim mehanizmom putem REST API-ja. Pri inicijalizaciji, instanca radnika konfigurira funkcije za dohvaćanje i zaključavanje aktivnosti (eng., *fetch_and_lock*), završavanje aktivnosti (eng., *complete_task*) te upravljanje neuspjesima (eng., *handle_failure*). Radnik može obrađivati više aktivnosti istovremeno, a implementiran je i mehanizam pretplate na određene teme (eng., *topics*) kako bi se omogućila specijalizirana obrada aktivnosti.

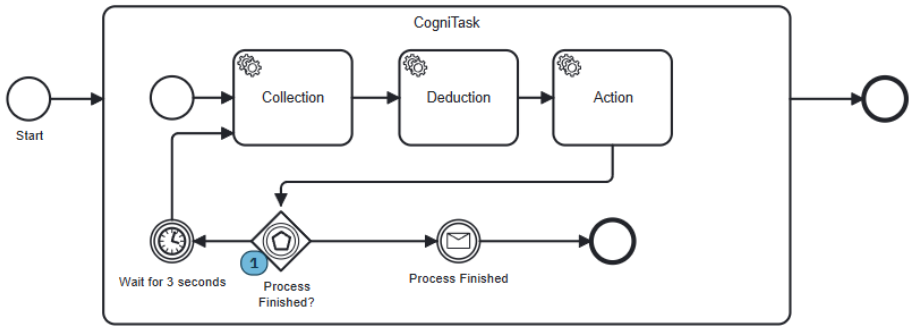
Tijekom izvršavanja (metoda *run*()), radnik (eng., *worker*) kontinuirano dohvaća aktivnosti iz Camunde. Kada aktivnost postane dostupna, prosljeđuje ju odgovarajućoj funkciji za obradu na temelju teme (eng., *topic*) kojoj pripada. Ako funkcija eksplicitno vrati podatke, oni se prosljeđuju natrag Camundi putem metode *complete_task()*. U slučaju neuspjeha, radnik poziva *handle_failure()*, pri čemu se definiraju detalji greške i određuje broj preostalih pokušaja ponovnog izvršavanja aktivnosti.

U skripti su implementirane tri ključne funkcije za obradu aktivnosti: *process_collection()*, *process_deduction()* i *process_action()*. Funkcija *process_collection()* odgovorna je za prikupljanje podataka i vraća poruku o uspješnom prikupljanju podataka. Funkcija *process_deduction()* prima rezultat prikupljenih podataka i temeljem njih provodi analizu ili predikciju, vraćajući odgovarajući rezultat. Funkcija *process_action()* koristi rezultate analize kako bi pokrenula odgovarajuće operativne radnje unutar sustava.

U glavnom dijelu skripte (*if __name__ == '__main__'*:), inicijalizira se radnik s definiranim URL-om Camunda REST API-ja i jedinstvenim identifikatorom. Radnik se potom pretplaćuje na tri teme (*collection, deduction, action*), pri čemu se svakoj od njih pridružuje odgovarajuća funkcija obrade podataka. Na kraju, metoda *run()* pokreće petlju koja omogućuje neprekidnu obradu vanjskih zadataka. Na slici 82 u nastavku se nalazi prikaz izvođenja *worker.py* skripte u terminalu.

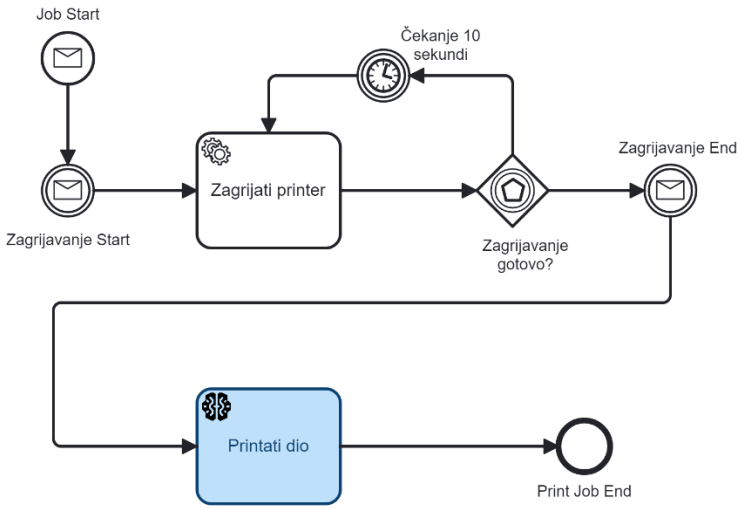


Slika 82 Prikaz izvođenja Python skripte worker.py

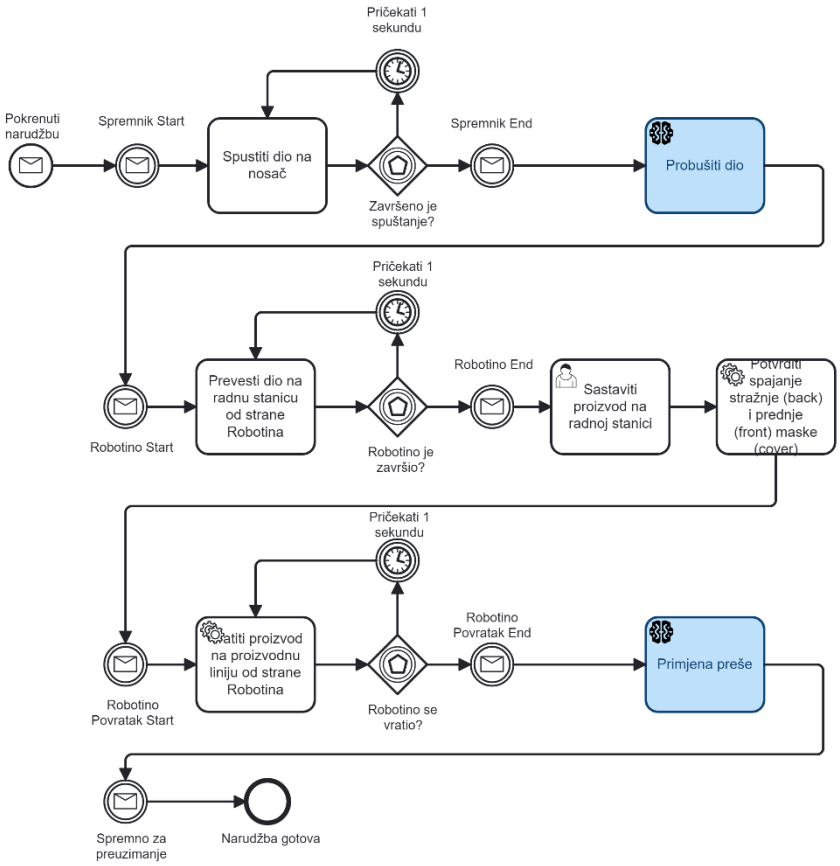


Slika 83 Izvođenje CogniTaska u Camundinoj platformi

CogniTask bi olakšao razvoj digitalnih blizanaca proizvodnih poslovnih procesa, koji uključuju element prediktivnog održavanja temeljem umjetne inteligencije. Ova nova vrsta aktivnosti omogućila bi izravnu integraciju modela strojnog učenja, poput neuronskih mreža, unutar proizvodnih poslovnih procesa, čime bi se značajno pojednostavila i ubrzala integracija prediktivnih analitičkih rješenja u BPMN dijagrame. Vanjski zadaci komuniciraju putem *topica* s digitalnim blizancem i to predstavlja napor za dizajnere poslovnih procesa koji često nisu stručnjaci za programiranje ili umjetnu inteligenciju. Prednost CogniTaska je, da uz svoju glavnu ulogu, omogućava labavu povezanost (eng., *loose coupling*) gdje dizajner procesa opisuje proces i specificira *topic* preko kojeg vanjski zadatak komunicira s glavnim procesom. Uvođenjem CogniTaska, dizajner bi mogao jednostavno povezati određeni korak u procesu s unaprijed definiranim modelom strojnog učenja (npr., predviđanje kvara opreme), poslati ulazne podatke, izvršiti predikciju i dobiti rezultat u realnom vremenu, sve unutar samog BPMN dijagrama, bez potrebe za dodatnom logikom u procesu. Time bi se omogućilo brže i intuitivnije kreiranje digitalnih blizanaca proizvodnih procesa, u kojima prediktivna analitika igra važnu ulogu u donošenju odluka. CogniTask postao bi tako središnja točka povezivanja između poslovnih procesa i sustava temeljenih na umjetnoj inteligenciji, čime bi alati za modeliranje poslovnih procesa postali prilagođeni suvremenim zahtjevima nadolazeće Industrije 5.0 i digitalnih blizanaca.
Na slikama 84 i 85 u nastavku se nalaze prikazi digitalnih blizanaca promatranih slučajeva korištenja u ovom istraživanju s predloženim proširenjem alata za modeliranje poslovnih procesa: CogniTaskom.



Slika 84 Digitalni blizanac proizvodnje drona sa simboličkim prikazom CogniTaska



Slika 85 Digitalni blizanac proizvodnje mobilnih uređaja sa simboličkim prikazom CogniTaska[11]

Uvođenje CogniTaska značajno poboljšava preglednost BPMN dijagrama, čime se povećava razumljivost modela. Kada bi takve funkcionalnosti bile podržane unutar alata poput Camunde, omogućila bi se učinkovitija izgradnja i modeliranje digitalnih blizanaca poslovnih procesa. Na temelju analize implementacije u Camundi, identificirani su ključni zahtjevi koje alati za modeliranje poslovnih procesa trebaju ispuniti kako bi podržali razvoj proizvodnih digitalnih blizanaca, a to su:

1. mogućnost povezivanja sa stvarnim procesima,
2. lančano povezivanje vanjskih sustava kroz procesne varijable
3. integracija prediktivnih modela.

Kako bi se to postignulo, prijedlog je da se uvede jedna uslužna aktivnost koja bi zapravo bila vanjska, a koja bi u sebi integrirala sve definirane zahtjeve. Novouvedena uslužna aktivnost mogla bi postati standardizirani dio svih alata za modeliranje poslovnih procesa, bez obzira na proizvođača.

[11] Ikona korištena za označavanje CogniTaska je slobodna za preuzimanje (MIT Licenca) sa stranice: https://www.svgrepo.com/svg/444899/ai-solid

17 Primjenjivost predložene metodike razvoja proizvodnog digitalnog blizanca

Razvoj proizvodnog digitalnog blizanca temelji se na sustavnom i iterativnom procesu koji, između ostalog, uključuje analizu, modeliranje, prediktivnu analitiku, integraciju s vanjskim sustavima te praćenje rada digitalnog blizanca u stvarnom proizvodnom okruženju. Predložena metodika za razvoj proizvodnih digitalnih blizanaca sastoji se od niza ključnih koraka, od kojih su oni, koji su implementirani u sklopu ovog istraživanja, naglašeni na slici 86 u nastavku. Implementacija metodike obuhvaćala je sve faze razvoja digitalnog blizanca, od analize i mapiranja proizvodnih poslovnih procesa do puštanja digitalnog blizanca u pogon i praćenja njegovog izvođenja.

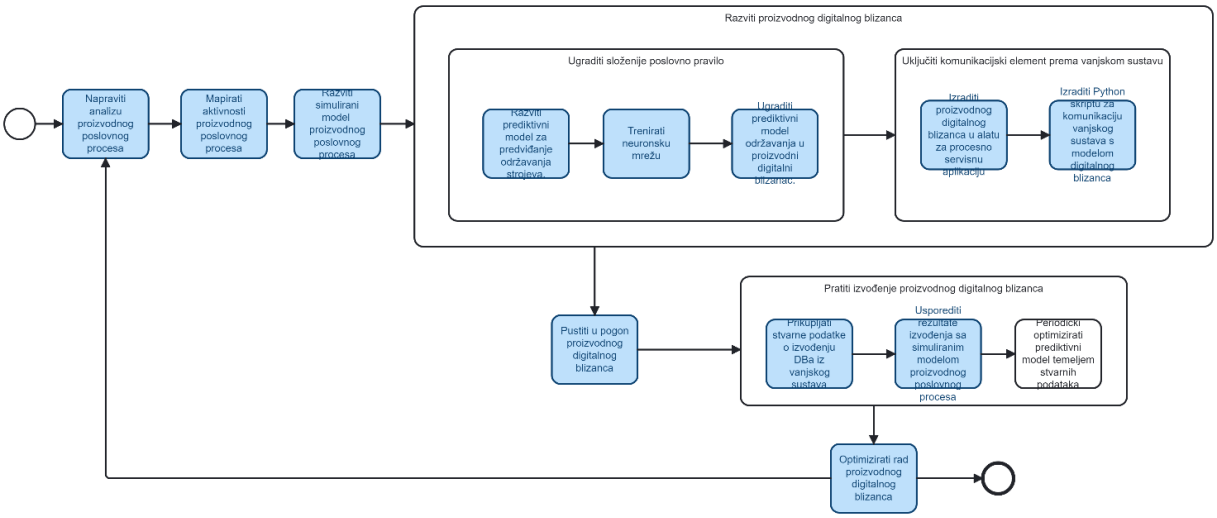
U okviru provedene analize, opisani su poslovni procesi dvaju proizvodnih sustava: proizvodnja dronova u Švicarskom Inovacijskom parku Biel/Bienne te Pametna tvornica na Fakultetu strojarstva i brodogradnje Sveučilišta u Zagrebu čime je postignut prvi korak predložene metodike. Ovi procesi su zatim modelirani u Camunda Modeleru, čime je ostvaren drugi korak metodike koji se odnosi na mapiranje aktivnosti proizvodnog poslovnog procesa. Izrađeni su i simulirani modeli poslovnih procesa u IBM WebSphere Business Modeleru kako bi se rezultati izvođenja proizvodnih digitalnih blizanaca mogli usporediti s rezultatima puštenih simulacija.

Značajan doprinos istraživanja je u implementaciji prediktivnog održavanja kao složenijeg poslovnog pravila unutar proizvodnog digitalnog blizanca. Razvijeni su prediktivni modeli bazirani na neuronskim mrežama, koji omogućuju rano otkrivanje potencijalnih kvarova na ključnim proizvodnim resursima, uključujući 3D printere, prešu i bušilicu. Ovi modeli su implementirani unutar digitalnih blizanaca u Camundi i povezani su sa sustavima praćenja stvarnog proizvodnog procesa putem Python skripti. Konkretno, u slučaju proizvodnje dronova, prediktivni model održavanja 3D printera oslanja se na dvije klasifikacijske neuronske mreže: binarni model za detekciju potencijalnog kvara i višeklasni model za klasifikaciju vrste kvara. S druge strane, kod proizvodnje mobilnih uređaja, razvijena je neuronska mreža s dvostrukim izlazom, koja istovremeno predviđa mogućnost kvara i tip kvara nad strojevima. Integracija prediktivnih modela u digitalne blizance pokazala se kao učinkovita metoda za unapređenje proizvodnih procesa, omogućujući pravovremeno prepoznavanje potencijalnih problema kako bi se smanjio zastoj u proizvodnji.

Osim implementacije prediktivnog održavanja, metodika je uključivala i razvoj komunikacijskih elemenata koji omogućuju povezivanje digitalnih blizanaca s vanjskim sustavima. Python skripte su korištene za periodičko dohvaćanje podataka iz baza podataka, praćenje statusa proizvodnih procesa te njihovu integraciju s Camundom. Implementirani mehanizmi osigurali su da digitalni blizanci kontinuirano reflektiraju stvarno stanje proizvodnje, čime je potvrđena njihova primjenjivost u stvarnom industrijskom okruženju. Nadalje, proizvodni digitalni blizanci su pušteni u pogon i testirani su, kako u stvarnim proizvodnim okruženjima, tako i u simuliranim uvjetima, čime je potvrđeno da modeli digitalnih blizanaca odražavaju realne uvjete poslovnih procesa.

Dio metodike odnosi se na usporedbu izvođenja digitalnih blizanaca sa simuliranim modelima poslovnih procesa. Analizom rezultata, utvrđeno je da između simuliranih modela i digitalnih blizanaca nema značajnih odstupanja u redoslijedu aktivnosti i ukupnom trajanju procesa, što potvrđuje da digitalni blizanci vjerno reproduciraju realne poslovne procese. Pritom su manja odstupanja uočena kod aktivnosti koje uključuju ljudsku intervenciju, što upućuje na razliku između tradicionalnih simuliranih modela i digitalnih blizanaca: dok simulirani modeli koriste statičke

parametre izvedene iz povijesnih podataka, digitalni blizanci omogućuju dinamičko praćenje i prilagodbu stvarnim uvjetima proizvodnje. Konačno, ako je to potrebno, proizvodne digitalne blizance moguće je optimizirati temeljem podataka prikupljenih tijekom njihovih izvođenja, čime je ostvarena posljednja faza metodike. Time je prikazano kako metodika razvoja proizvodnih digitalnih blizanaca nije samo teorijski okvir, već praktično primjenjiv pristup koji omogućuje razvoj i implementaciju digitalnih blizanaca u realnim proizvodnim sustavima.



Slika 86 Metodika razvoja proizvodnog digitalnog blizanca po implementiranim koracima

Iako korak "Periodički optimizirati prediktivni model temeljem stvarnih podataka" nije implementiran u ovoj fazi istraživanja, predložena metodika omogućuje njegovu buduću integraciju. Kada digitalni blizanac bude duže u produkciji i kada će biti moguće prikupiti dovoljnu količinu novih podataka, prediktivni model moći će se iterativno poboljšavati kako bi bolje reflektirao promjene u proizvodnom sustavu. Primarni cilj istraživanja bio je testirati funkcionalnost predloženog pristupa digitalnim blizancima i integraciju prediktivnih modela u proizvodne poslovne procese, a ne nužno iterativno poboljšavanje prediktivnog modela kroz više ciklusa učenja. Istraživanje je bilo usmjereno na inicijalnu implementaciju prediktivnih modela, njihovu integraciju u proizvodne digitalne blizance, razvoj proizvodnih digitalnih blizanaca i provjeru njihove točnosti.

Iako se predložena metodika pokazala primjenjivom u proizvodnim okruženjima koja uključuju jasno definirane poslovne procese, dostupne modele podataka i mogućnost integracije s vanjskim sustavima, važno je istaknuti da metodika nije nužno primjenjiva na sve tipove proizvodnih i poslovnih sustava. Metodika je posebno prilagođena proizvodnim sustavima u kojima je moguće identificirati tokove aktivnosti, evidentirati relevantne senzorske podatke te ih povezati s poslovnim pravilima i prediktivnim modelima. Ograničenja proizlaze kako iz proizvodnog, tako i iz poslovnog konteksta. Kompleksnost poslovnih sustava i nepostojanje standardizirane infrastrukture može otežati implementaciju i održavanje digitalnog blizanca. Metodika stoga nije nužno primjenjiva na poslovne sustave koji se temelje na nestrukturiranim podacima i koji nemaju stabilne poslovne procese. U tom smislu, klasa problema na koju se metodika primjenjuje su strukturirani proizvodni procesi s mogućnošću prikupljanja i korištenja podataka u svrhu podrške donošenju odluka, uz tehničke preduvjete za povezivanje sustava u realnom vremenu.

18 Znanstveni doprinos istraživanja

Ideja ovog istraživanja je bila predložiti metodiku razvoja proizvodnih digitalnih blizanaca jer je to prepoznato kao istraživački jaz koji nije dovoljno pokriven znanstveno, istraživački, a ni stručno. Digitalni blizanci su širok pojam i primjenjivi su danas u velikoj sferi života ljudi, istraživača i projektanata. Od toga da se rade digitalni blizanci uređaja, proizvoda, gradova, do toga da se može napraviti digitalni blizanac proizvodnog poslovnog procesa. Međutim, nigdje nije opisano kako bi izgledala metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa i kako bi on izgledao u stvarnosti. Razmišljajući kako bi se razvio digitalni blizanac proizvodnje, došlo je do otkrića novog jaza, a to je da li postojeći načini modeliranja poslovnih procesa uopće pokrivaju izradu modela proizvodnog digitalnog blizanca. Razvoj poslovnih procesa je napredovao i ponekad se ne možemo oslanjati na postojeća i davno utemeljena rješenja već trebamo pratiti trendove razvoja i, sukladno tome, potrebe tržišta kojima se trebamo prilagoditi. Tijekom faze razvoja proizvodnih digitalnih blizanca, uočeni su zahtjevi koje postojeći alati za modeliranje poslovnih procesa trebaju zadovoljiti kako bi podržali izradu proizvodnih digitalnih blizanaca, naročito onih koji u sebi implementiraju modele prediktivnog održavanja. Postavlja se pitanje zašto baš proizvodnih, zato što je pri razvoju proizvoda najbitnije pratiti tijek izvođenja procesa kako bi se na vrijeme uočile moguće poteškoće u proizvodnji i pravovremeno reagiralo kroz donošenje optimalnih odluka u pravom trenutku. Optimalne odluke su u ovom slučaju operativne odluke koje se donose prije ispada i mogućih problema u proizvodnji kako bi se iste izbjegle, a proizvodnja se neometano nastavila. S obzirom na današnju sve veću uključenost elementa umjetne inteligencije i predviđanja događaja u svim sferama života, ovaj element je uočen kao doprinos kojeg treba uključiti u proizvodni proces, odnosno, implementirati u proizvodnog digitalnog blizanca.

Ciljevi ovog istraživanja bili su:

(1) Oblikovati metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa proširenjem postojeće metodike modeliranja poslovnih procesa.

(2) *Razviti digitalnog blizanca proizvodnog poslovnog procesa primjenom metodike za oblikovanje digitalnih blizanaca proizvodnih poslovnih procesa.*

Prvi istraživački cilj je postignut u poglavlju „Metodika razvoja digitalnog blizanca proizvodnog poslovnog procesa*“* gdje je temeljem postojećih metodika i ranije definiranih koraka izrade proizvodnog digitalnog blizanca oblikovana metodika za razvoj digitalnih blizanca proizvodnih poslovnih procesa.

Drugi istraživački cilj je postignut kroz poglavlje „Digitalni blizanci napravljeni prema novoj metodici razvoja proizvodnih digitalnih blizanaca“ gdje su kroz dva slučaja korištenja prikazani proizvodni digitalni blizanci s ugrađenim modelom prediktivnog održavanja opreme.
Istraživačko pitanje „*Kako dizajnirati metodiku za razvoj digitalnih blizanaca proizvodnih poslovnih procesa?*“ odgovoreno je u poglavlju „Koraci razvoja digitalnog blizanca proizvodnog poslovnog procesa“ gdje su opisani koraci razvoja proizvodnog digitalnog blizanca i napravljen je uvod u metodiku razvoja.

Hipoteza „*Digitalni blizanac proizvodnog poslovnog procesa razvijen proširenom metodikom modeliranja poslovnih procesa emulira realni proizvodni poslovni proces kako se on odvija u stvarnosti.*“ potvrđena je putem simulacijskog eksperimenta i rezultati usporedbe simuliranog modela i stvarnog modela proizvodnog digitalnog blizanca pokazuju kako nema značajnog odstupanja u izvođenju digitalnog blizanca proizvodnog poslovnog procesa s prediktivnim održavanjem i simuliranog modela poslovnog procesa napravljenog temeljem povijesnih podataka. Rezultati toga vidljivi su u poglavlju „Usporedba simuliranih modela poslovnih procesa s digitalnim blizancima poslovnih procesa“.

Kroz tri faze simulacijskog eksperimenta ovo istraživanje je prikazalo razvoj, testiranje i validaciju digitalnih blizanaca, s naglaskom na prediktivno održavanje opreme u proizvodnim sustavima. Način kako se ovo istraživanje uklopilo unutar faza dizajniranja simulacijskog eksperimenta prema (Baeldung, 2024) je:

1. U fazi modeliranja, izradio se digitalni blizanac proizvodnog poslovnog procesa na temelju prikupljenih podataka o proizvodnim procesima i opremi u proizvodnim sustavima. Za izradu modela proizvodnih digitalnih blizanaca se koristila BPMN notacija u Camundi uz proširenje Python skriptama, dok su se prediktivni modeli razvili u obliku neuronskih mreža u Pythonu s TensorFlowom. Modeli su se validirali i verificirali usporedbom predviđenih ishoda digitalnog blizanca s realnim podacima dobivenima iz proizvodnog sustava. Također, provelo se treniranje neuronskih mreža i njihovo testiranje korištenjem otvorenog skupa podataka.
2. U eksperimentalnoj fazi, definirali su se eksperimentalni uvjeti i provele su se simulacije kako bi se testirala učinkovitost digitalnih blizanaca. Eksperimenti su se ponavljali višestrukim pokretanjem simulacija pod istim inicijalnim uvjetima, pri čemu su se kontinuirano prikupljali podaci o stanju procesa i rezultatima neuronskih mreža. Korištenjem Camunde i Python skripti proveo se *polling* baze podataka, omogućujući *real-time* ažuriranje modela prema promjenama u procesu.
3. U fazi analize podataka, analizirali su se rezultati simulacija i predikcija digitalnih blizanaca. Evaluirala se preciznost modela neuronskih mreža na testnim podacima te uspješnost prediktivnog održavanja. Izrađeni su simulacijski modeli u IBM WebSphere-u, koji su zatim korišteni za usporedbu s rezultatima izvođenja proizvodnih digitalnih blizanaca. Analizirane su razlike između izvođenja digitalnih blizanaca i ishoda simulacijskih modela kako bi se ocijenila točnost i učinkovitost razvijenog sustava.

Znanstveni doprinos ovog istraživanja vidi se u više međusobno povezanih tehnoloških i metodoloških razina, koje zajedno čine sveobuhvatan pristup razvoju proizvodnih digitalnih blizanaca temeljenih na stvarnim poslovnim procesima. U okviru istraživanja, razvijen je cjelovit metodološki okvir koji objedinjuje tehnike iz područja simulacijskog modeliranja, prediktivne analitike i automatizacije poslovnih procesa.

Jedan od doprinosa je izgradnja vlastite baze podataka temeljene na stvarnim i simuliranim podacima. Baza je koncipirana kao vremenska (temporalna) baza transakcijskog tipa, u kojoj se pohranjuju podaci o vremenu trajanja procesa, parametrima strojeva, statusima aktivnosti i ishodima izvođenja poslovnih operacija. Baza omogućuje analiziranje tokova i uzoraka ponašanja unutar proizvodnih procesa, ali i služi kao temelj za treniranje prediktivnog modela.

U istraživanju je razvijen i istreniran prediktivni model temeljen na višeslojnoj neuronskoj mreži, čija je svrha predviđanje održavanja opreme u stvarnom vremenu. Ovaj model integriran je u digitalnog blizanca pomoću poslovnih pravila u Camunda BPMN okruženju, čime je omogućena automatizirana reakcija na potencijalne kvarove u poslovnom procesu.

Poseban znanstveni doprinos ogleda se i u razvoju vlastitog softverskog rješenja koje povezuje Camundu, kao alat za upravljanje i modeliranje poslovnih procesa, s programskim komponentama napisanima u Pythonu. Python skripte razvijene su za obradu podataka, aktivaciju vanjskih usluga te izvršavanje predikcije, što predstavlja primjer integracije poslovne logike i napredne analitike unutar istog digitalnog modela.

Nadalje, doprinos istraživanja uključuje provedbu simulacijskog modeliranja proizvodnog procesa pomoću IBM WebSphere Business Modelera, čime se validira ponašanje digitalnog blizanca kroz usporedbu sa stvarnim povijesnim podacima.

Ključan znanstveni doprinos je predložena metodika razvoja digitalnog blizanaca proizvodnog poslovnog procesa koja je primjenjiva na proizvodna poslovna poduzeća u kojima su jasno definirani poslovni procesi te u kojima postoji mogućnost povezivanja s vanjskim sustavima. Važno je istaknuti da je predložena metodika testirana kroz dva slučaja korištenja u stvarnim organizacijama, čime se potvrđuje njezina primjenjivost u realnim uvjetima. Time se ne potvrđuje samo funkcionalnost razvijenog softverskog rješenja, nego i znanstvena vrijednost metodike koja objedinjuje pristupe iz različitih domena: razvoja softvera, simulacije i upravljanja poslovnim procesima. Korake razvoja digitalnog blizanca može tako pratiti proizvodno poduzeće bez obzira da li uključuje element umjetne inteligencije u proizvodnju ili ne. Ako proizvodno poduzeće ne uključuje element umjetne inteligencije u proizvodnju, prilikom razvoja proizvodnog digitalnog blizanca taj korak će zaobići i nastaviti s daljnjim koracima. Međutim, prijedlog kako unaprijediti postojeće alate za modeliranje poslovnih procesa je dio koji ostaje i daje velik značaj za budućnost kada se očekuje veliki skok uključenosti umjetne inteligencije u proizvodnju i to je bitno za današnje vrijeme kada se umjetna inteligencija ne može zaobići.

Konačno, predložena vanjska aktivnost, CogniTask, predstavlja posljednji znanstveni doprinos jer bi omogućila integraciju modela strojnog učenja unutar BPMN dijagrama, čime bi se otklonile postojeće prepreke u primjeni prediktivne analitike u proizvodnim poslovnim procesima. Trenutno, implementacija prediktivnih modela zahtijeva vanjske skripte i ručne API integracije, što otežava korištenje ovih metoda dizajnerima poslovnih procesa koji nisu specijalizirani za umjetnu inteligenciju ili programiranje. CogniTask bi riješio ovaj problem pružajući intuitivni koncept koji bi omogućio jednostavno povezivanje prediktivnih modela s poslovnim procesima, automatizirano slanje podataka, izvođenje predikcija i donošenje odluka u realnom vremenu. Ovaj doprinos je značajan jer obogaćuje predloženu metodiku razvoja digitalnih blizanaca proizvodnih procesa, čineći ih inteligentnijima i prilagođenima potrebama industrijske digitalizacije. Važnost ovog znanstvenog doprinosa očituje se u unapređenju alata za modeliranje poslovnih procesa, koji ne podržavaju integraciju prediktivnih modela na standardiziran način. Kroz analizu implementacije u Camundi identificirani su ključni zahtjevi za razvoj proizvodnih digitalnih blizanaca u kontekstu prediktivnog održavanja, uključujući povezivanje s fizičkim procesima, obradu odgovora iz vanjskih sustava i integraciju prediktivnih modela. Predloženo rješenje, CogniTask, bi objedinio sve ove zahtjeve u jednu cjelinu, čime bi se omogućila lakša i brža primjena prediktivne analitike u poslovnim

procesima. Ovaj znanstveni doprinos pomaže standardizaciji modeliranja proizvodnih digitalnih blizanaca i osigurava da alati za modeliranje poslovnih procesa budu prilagođeni zahtjevima nadolazeće Industrije 5.0, gdje inteligentna automatizacija igra ključnu ulogu.

19 Zaključak

Integracija proizvodnih digitalnih blizanaca u poslovne procese zahtijeva proširenje funkcionalnosti alata za modeliranje poslovnih procesa, neovisno o konkretnoj platformi ili proizvođaču. Bez obzira koristi li se Camunda, Flowable, Bizagi ili neki drugi alat, ključno je da svi podržavaju jedinstvene standardizirane mehanizme za povezivanje s fizičkim procesima i sustavima temeljenima na umjetnoj inteligenciji. Konkretno, alati trebaju omogućiti jednostavno konfiguriranje ulaznih i izlaznih konektora za povezivanje s vanjskim sustavima putem URL-a i REST API-ja, kao i mogućnost obrade povratnih podataka iz tih sustava kroz procesne varijable. Dodatno, alati bi trebali uključivati specijaliziranu aktivnost za integraciju modela strojnog učenja koja bi omogućila lakše povezivanje poslovnih procesa s prediktivnim analitičkim rješenjima. Samo kroz standardizaciju ovih elemenata na razini svih alata, poslovni procesi će moći u potpunosti podržati razvoj i operativno korištenje proizvodnih digitalnih blizanaca, neovisno o izboru specifične platforme. Ovaj pristup ne samo da unapređuje trenutno stanje modeliranja poslovnih procesa, već osigurava dugoročnu prilagodljivost sustava u kontekstu Industrije 5.0 i sve većeg uključivanja inteligentne automatizacije u poslovne operacije.

Cijeli sustav predstavljen u ovom istraživanju oslanja se na komunikaciju između digitalnog blizanca, prediktivnog modela i stvarnog sustava, omogućujući kontinuirano poboljšanje modela digitalnog blizanca i njegovu prilagodbu u stvarnom vremenu. Kroz prezentiranu metodiku, proizvodni digitalni blizanac replicira stvarni proizvodni poslovni proces, predviđa potencijalne kvarove te optimizira operacije na temelju analize podataka iz stvarnog svijeta.
Predložena metodika razvoja digitalnih blizanaca proizvodnih poslovnih procesa proširuje postojeće načine modeliranja poslovnih procesa, osiguravajući njihovu prilagodbu zahtjevima industrije. Implementacija prediktivnog održavanja unutar digitalnog blizanca omogućuje pravovremeno donošenje operativnih odluka, čime se povećava učinkovitost i pouzdanost proizvodnih sustava. Kroz dva slučaja korištenja prikazana je primjenjivost metodike u proizvodnoj industriji.

Razvijeni koncept proizvodnog digitalnog blizanca potvrđen je kroz simulacijski eksperiment, pri čemu su dobiveni rezultati pokazali da predložena metodika omogućuje realističnu emulaciju proizvodnih procesa i učinkovito prediktivno održavanje. Time se otvara prostor za daljnja istraživanja i unapređenja, primjerice u smjeru širenja metodike na druge industrijske sektore, poput logistike, zdravstva ili energetike, gdje digitalni blizanci također mogu imati značajan utjecaj na optimizaciju poslovanja. Dodatno, buduća istraživanja mogu biti usmjerena na dublju integraciju drugih, različitih metoda umjetne inteligencije unutar digitalnih blizanaca. Sustavna evaluacija dugoročnih učinaka digitalnih blizanaca na organizacijsku učinkovitost, troškove i održivost poslovanja može također biti zanimljiv pravac daljnjih istraživanja.

20 Popis slika

21 Popis tablica

22 Literatura

Aazam, M., Zeadally, S., & Harras, K. A. (2018). Fog computing architecture, evaluation, and future research directions. *IEEE communications magazine*, *56(5)*, 46–52.

Abouzid, I. & Saidi, R. (2023). Digital twin implementation approach in supply chain processes. *Scientific African 21*. https://doi.org/10.1016/j.sciaf.2023.e01821

Agrawal, A., Thiel, R., Jain, P., Singh, V., & Fischer, M. (2023). Digital Twin: Where do humans fit in? *Automation in Construction, Volume 148*. https://doi.org/10.1016/j.autcon.2023.104749

AI4I 2020 Predictive Maintenance Dataset. (2020). UC Irvine, Machine Learning Repository. https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset

Arnold, L., Breitmayer, M., & Reichert, M. (2025). A Web-Based Modelling Tool for Object-Centric Business Processes. *Lecture Notes in Business Information Processing , 537 LNBIP*, 267–275.

Asana. (2024). *7 types of process improvement methodologies you should know about*. https://asana.com/resources/process-improvement-methodologies

Aurich, J.C., Mertes, J., Wagner, M., Schmitz, M., Klar, M., & Ravani, B. (2025). Digital twins in manufacturing: A taxonomy for manufacturing applications. *Digital Twin, 2(1)*. https://doi.org/10.1080/27525783.2025.2496645

Baeldung. (2024). *How to Design Simulation Experiments?* https://www.baeldung.com/cs/research-design-simulation-experiments

Beke, Á.K., Gyürkés, M., Nagy, Z.K., Marosi, G., & Farkas, A. (2021). Digital twin of low dosage continuous powder blending – Artificial neural networks and residence time distribution models. *European Journal of Pharmaceutics and Biopharmaceutics*, *169*, 64–77.

Bocciarelli, P., D'Ambrogio, A., & Panetti, T. (2023). A Model Based Framework for IoT-Aware Business Process Management. *Future Internet 15(2), 50*.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 13–16.

Botin-Sanabria, D.M., Mihaita, A.-S., Peimbert-Garcia, R.E., Ramirez-Moreno, M.A., Ramirez-Mendoza, R.A., & Lozoya-Santos, J. (2022). Digital Twin Technology Challenges and Applications: A Comprehensive Review. *Remote Sens. 2022, 14(6)*. https://doi.org/10.3390/rs14061335

Brucherseifer, E., Marquard, M., Hellmann, M., & Tundis, A. (2024). A Data Taxonomy Towards the Applicability of the Digital Twin Conceptual Framework in Disaster Management. *arXiv:2503.00076v1 [cs.CY]*. https://arxiv.org/abs/2503.00076

Camunda: The Universal Process Orchestrator. (bez dat.). https://camunda.com/

Cao, X., Yao, M., Zhang, Y., Hu, X., & Wu, C. (2024). Digital Twin Modeling and Simulation Optimization of Transmission Front and Middle Case Assembly Line. *Computer Modeling in Engineering & Sciences*. https://doi.org/10.32604/cmes.2023.030773

Corradini, F., Pettinari, S., Re, B., Rossi, L., & Tiezzi, F. (2023). Executable Digital Process Twins: Towards the Enhancement of Process-Driven Systems. *Big data and cognitive computing*. https://doi.org/ 10.3390/bdcc7030139

Dashkina, A., Khalyapina, L., Kobicheva, A., Malykhina, G., & Tarkhov, D. (2020). Neural network modeling as a method for creating digital twins: From Industry 4.0 to Industry 4.1. *ACM International Conference Proceeding Series*.

der Landwehr, M.A., Trott, M., & von Viebahn, C. (2021). Waste of time and money? Constructing an applicability framework for organizational use of simulation studies and digital twins. *International Conference on Information Systems, ICIS 2020 - Making Digital Inclusive: Blending the Local and the Global*.

Digital Twin Consortium. (2024). *Digital Twins: A framework that unlocks business value and enables digital transformation at scale*. https://www.digitaltwinconsortium.org/2024/04/digital-twins-a-framework-that-unlocks-business-value-and-enables-digital-transformation-at-scale/

Ding, K., Fan, L., & He, Ch. (2024). Orchestration constraints BlockChained smart contract and integrated digital twins manufacturing systems: ManuChain4.0-based application paradigm and case study. *Journal of Manufacturing Systems, 74*, 606–632. https://doi.org/10.1016/j.jmsy.2024.04.014

Dumas, M. (2021). Constructing digital twins for accurate and reliable what-if business process analysis. *CEUR Workshop Proceedings 2938*, 23–27.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013). *Fundamentals of Business Process Management*. Springer-Verlag Berlin Heidelberg.

Dymitrowski, A. & Mielcarek, P. (2021). Business model innovation based on new technologies and its influence on a company's competitive advantage. *Journal of Theoretical and Applied Electronic Commerce Research, 16(6)*, 2110–2128.

Fakultet strojarstva i brodogradnje. (2022). *Pametna proizvodnja i Industrija 4.0*.

Fakultet strojarstva i brodogradnje. (2024). *O Fakultetu strojarstva i brodogradnje*. https://www.fsb.unizg.hr/index.php?fsbonline&o_fakultetu

Fuhrländer-Völker, D., Lindner, M., Von Elling, M., Frieß, T., Karnapp, S., & Weigold, M. (2025). Method for the development and application of digital twins in manufacturing. *Production Engineering , 012001*.

Galera-Zarco, C. & Papadonikolaki, E. (2023). DIGITAL TWIN-BASED SERVICES: A TAXONOMY (Extended Abstract). *International Conference on AI and the Digital Economy (CADE 2023)*.

Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly: Management Information Systems, 28(1),* 75–105.

Hielscher, T., Khalil, S., Virgona, N., & Hadigheh, S.A. (2023). A neural network based digital twin model for the structural health monitoring of reinforced concrete bridges. *Structures, 57*(105248).

IBM. (bez dat.). *Download WebSphere Business Modeler Advanced V7.0*. https://www.ibm.com/support/pages/download-websphere-business-modeler-advanced-v70

IBM. (2025). *What is business process modeling and notation (BPMN)?* https://www.ibm.com/think/topics/bpmn

Iizuka, K., Okawada, T., Shime, Y., Iizuka, Y., & Suematsu, C. (2014). Inter-organizational process flow adjustment: Applying multi-layered process flow adjustment tool. *Proceedings - 2014 IIAI 3rd International Conference on Advanced Applied Informatics, IIAI-AAI 2014*, 867–872.

Indeed Editorial Team. (2025). *Production Process: Definition and Types for Businesses To Use*. https://www.indeed.com/career-advice/career-development/production-process

interfacing. (2025). *An Introduction to BPMN 2.0 Symbology*. https://interfacing.com/bpmn-2-0-symbology

Jiang, Y. (2021). Intelligent Building Construction Management Based on BIM Digital Twin. *Hindawi Computational Intelligence and Neuroscience Volume 2021*. https://doi.org/10.1155/2021/4979249

Jin, G., Jin, G., & Huo, H. (2022). Selection of Business Process Modeling Tool with the Application of Fuzzy DEMATEL and TOPSIS Method. *Axioms, 11(11), 601*.

Jinzhi, L., Zhaorui, Y., Xiaochen, Z., Jian, W., & Dimitris, K. (2022). Exploring the concept of Cognitive Digital Twin from model-based systems engineering perspective. *The International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-022-09610-5

Kaggle. (2020). *Machine Predictive Maintenance Classification*. Kaggle. https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification/data

Križanić, S. (2020). Educational data mining using cluster analysis and decision tree technique: A case study. *International Journal of Engineering Business Management , 12*.

Križanić, S. & Vrček, N. (2023). Business process modeling and digital twins: A literature review. *Proceedings of the Central European Conference on Information and Intelligent Systems, 34th CECIIS*, 145–151.

Križanić, S. & Vrček, N. (2024). Development of a Process Digital Twin in Camunda Modeler. *TEM Journal , 13(4)*, 3315–3323.

Kuechler, B. & Vaishnavi, V. (2008). Theory Development in Design Science Research: Anatomy of a research project. *Proceedings of the 3rd International Conference on Design Science Research in Information Systems and Technology, DESRIST 2008*, 1–15.

Laurene Fausett. (1994). *Fundamentals of neural networks: Architectures, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458.

Liu, Y., Yang, C., Jiang, L., Xie, S., & Zhang, Y. (2020). Intelligent edge computing for IoT-based energy management in smart cities. *IEEE Network, 34(2)*, 44–51.

Lu, Y., Zhao, G., Xu, C., Imran, M., Yu, K., & Rodrigues, J.J.P.C. (2023). A Framework for Digital Twin-Based Deterministic Communication in Satellite Time Sensitive Networks. *IEEE International Conference on Communications (ICC): SAC Satellite and Space Communications Track*. https://doi.org/10.1109/ICC45041.2023.10279611

Lüftenegger, E. & Softic, S. (2020). SentiProMo: A Sentiment Analysis-Enabled Social Business Process Modeling Tool. *Lecture Notes in Business Information Processing , 397*, 83–89.

Lugaresi, G. & Matta, A. (2021). Automated digital twins generation for manufacturing systems: A case study. *IFAC-PapersOnLine, 54(1)*, 749–754.

Mallek-Daclin, S., Chaabaoui, L., Daclin, N., Rabah, S., & Zacharewicz, G. (2022). Planification model-based process discovering. *21st International Conference on Modeling and Applied Simulation, MAS 2022*.

Mallek-Daclin, S., Daclin, N., Rabah, S., & Zacharewicz, G. (2023). Product Development Plan Monitoring: Towards a Business Process Digital Twin. *IFAC-PapersOnLine , 56(2)*, 11894–11899.

Marmolejo-Saucedo, J.A. (2020). Design and Development of Digital Twins: A Case Study in Supply Chains. *Mobile Networks and Applications, 25(6)*, 2141–2160.

McKinsey&Company. (2024). *What is digital-twin technology?* https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-digital-twin-technology?cid=omcknsl-eml-nsl--mck-ext-----&hlkid=e32ea8a0d22f4081afc482d121b924a0&hctky=14794449&hdpid=dc4ac800-58fd-49b5-84f0-b7e812fd5700

Medeiros, A.L., De Sá Sousa, H.P., Santoro, F.M., Batista, T.V., & Da Silva José, H.S. (2017). Evolving Oryx into an aspect-oriented business process modelling tool. *International Journal of Business Process Integration and Management , 8(4),* 223–242.

Meierhofer, J., West, S., Rapaccini, M., & Barbieri, C. (2020). The Digital Twin as a Service Enabler: From the Service Ecosystem to the Simulation Model. *Lecture Notes in Business Information Processing 377 LNBIP,* 347–359.

Meziani, R. & Saleh, I. (2011). Towards a Collaborative Business Process Management Methodology. *International Conference on Multimedia Computing and Systems - Proceedings*.

Nast, B., Reiz, A., Sandkuhl, K., & Stirna, J. (2022). Integrating Organizational and Technological Aspects of Digital Twin Engineering. *CEUR Workshop Proceedings*, *3298*.

Neufeld, N. & Deo, B. (2021). Organizational fit and business process management implementation. *IISE Annual Conference and Expo 2021*, 548–553.

Node-RED. (2024). *Low-code programming for event-driven applications*. https://nodered.org/about/

Padmanabhuni, S., Ganesh, J., & Moitra, D. (2004). Web services, grid computing, and business process management: Exploiting complementarities for business agility. *Proceedings - IEEE International Conference on Web Services*, 666–673.

Palchunov, D. & Vaganova, A. (2021). Methods for Developing Digital Twins of Roles Based on Semantic Domain-Specific Languages. *International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices, EDM, 2021-June,* 515–519.

Park, G. & Van Der Aalst, W.M.P. (2021). Realizing A Digital Twin of An Organization Using Action-oriented Process Mining. *Van Der Aalst, W.M.P. Proceedings - 2021 3rd International Conference on Process Mining, ICPM,* 104–111.

Popa, C. & Goia, I. (2024). THE MODELLING OF CARGO TRANSSHIPMENT OPERATIONS USING THE BUSINESS PROCESS MODELLING TOOLS. *Scientific Journal of Silesian University of Technology. Series Transport , 124,* 157–169.

Raska, P., Ulrych, Z., & Malaga, M. (2021). Data Reduction of Digital Twin Simulation Experiments Using Different Optimisation Methods. *Applied Sciences 2021, 11, 7315*. https://doi.org/10.3390/app11167315

Raul Rojas. (1996). *Neural Networks, A Systematic Introduction*. Springer-Verlag Berlin Heidelberg.

Reijers, H.A., Freytag, T., Mendling, J., & Eckleder, A. (2011). Syntax highlighting in business process models. *Decision Support Systems , 51(3),* 339–349.

Romero, D., Von Cieminski, G., Wuest, T., (...), Alfnes, E., & Rudberg, M. (2021). Advances in Production Management Systems: Issues, Trends, and Vision Towards 2030. *IFIP Advances in Information and Communication Technology 600,* 194–221.

Saif, W., RazaviAlavi, SR., & Kassem, M. (2024). Construction digital twin: A taxonomy and analysis of the application-technology-data triad. *Automation in Construction 167:105715*. https://doi.org/10.1016/j.autcon.2024.105715

Shaw, M. (2003). *Writing Good Software Engineering Research Papers*. 726–736.

Simon Haykin. (2009). *Neural Networks and Learning Machines*. Pearson Education Inc., Upper Saddle River.

Sreedharan, S., Ramachandran, M., & Ramesh, D. (2025). Harnessing digital twins and industrial-IoT for cutting-edge mining automation: A methodological and technology assessment prototype. *Computers & Industrial Engineering*. https://doi.org/10.1016/j.cie.2025.110871

SUMEREDER, A. & WOITSCH, R. (2022). DIGITIZATION PRINCIPLES FOR APPLICATION SCENARIOS TOWARDS DIGITAL TWINS OF ORGANIZATIONS. *The 8th European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS Congress 2022*. https://doi.org/10.23967/eccomas.2022.091

Swiss Smart Factory. (bez dat.). https://www.sipbb.ch/en/forschung/swiss-smart-factory/

Switzerland Innovation Park Biel/Bienne. (bez dat.). https://www.sipbb.ch/en/

Szelągowski, M. & Berniak-Woźny, J. (2024). BPM challenges, limitations and future development directions – a systematic literature review. *Business Process Management Journal*, *30(2),* 505–557.

Švogor, Ivan. (2016). A framework for Allocation of Software Components onto a Heterogeneous Computing System. *University of Zagreb, Faculty of Organization and Informatics Varaždin, Doctoral thesis*. https://repozitorij.foi.unizg.hr/islandora/object/foi:526

Talkhestani, B.A., Jung, T., Lindemann, B., Sahlab, N., Jazdi, N., Schloegl, W., & Weyrich, M. (2019). An architecture of an Intelligent Digital Twin in a Cyber-Physical Production

System. *Automatisierungstechnik 2019; 67(9)*, 762–782. https://doi.org/10.1515/auto-2019-0039

Tang, W., Shen, B., & Chen, C. (2009). CuteFlow: A refined Modeling-Driven business process designer. *2009 WRI World Congress on Software Engineering, WCSE 2009*, *4*, 300–304.

Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., & Nee, A. Y. (2018). Digital twin-driven product design framework. *International Journal of Production Research*, *57(12)*.

TechTarget. (2023). *What is a digital twin and how does it work?*

Thabet, R., Lamine, E., Boufaied, A., Korbaa, O., & Pingaud, H. (2018). Towards a risk-aware business process modelling tool using the ADOxx platform. *Lecture Notes in Business Information Processing , 316*, 235–248.

Thabet, R., Lamine, E., & Pingaud, H. (2021). Development of a Risk-aware Business Process Modeling Tool for Healthcare processes. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*.

Tomaz, L.F.C., Rodrigues Nt, J.A., Souza, J.M., & Xexeo, G.B. (2011). Bringing knowledge into recommendation systems. *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2011*, 246–252.

Torres, J., San-Mateos, R., Lasarte, N., Mediavilla, A., Sagarna, M., & León, I. (2024). Building Digital Twins to Overcome Digitalization Barriers for Automating Construction Site Management. *Buildings, 14*. https://doi.org/10.3390/ buildings14072238

Ubaid, Alaa M & Dweiri, Fikri T. (2020). Business process management (BPM): Terminologies and methodologies unified. *International Journal of System Assurance Engineering and Management , 11(6)*, 1046–1064.

Uhlenkamp, J.-F., Hauge, J.B., Broda, E., & Thoben, K.-D. (2022). Digital Twins: A Maturity Model for Their Classification and Evaluation. *IEEE Access*. https://doi.org/10.1109/ACCESS.2022.3186353

Vaishnavi, V. & Kuechler, B. (2004). Design Science Research in Information Systems. *Computer Information Systems Department, Georgia State University*.

Van der Aalst, W.M.P. (2022). Six Levels of Autonomous Process Execution Management (APEM). *arXiv:2204.11328, Cornell University*. https://doi.org/10.48550/arXiv.2204.11328

Van der Valk, H., Haße, H., Möller, F., Arbter, M., Henning, J.-C., & Otto, B. (2020). A Taxonomy of Digital Twins. *26th Americas Conference on Information Systems (AMCIS)At: Salt Lake City, USA*.

Vaskovsky, A.M., Chvanova, M.S., & Rebezov, M.B. (2020). Creation of digital twins of neural network technology of personalization of food products for diabetics. *Conference*

Proceedings - 4th Scientific School on Dynamics of Complex Networks and their Application in Intellectual Robotics, DCNAIR 2020, 251–253.

Verma, S., Kawamoto, Y., Fadlullah, Z. M., Nishiyama, H., & Kato, N. (2017). A survey on network methodologies for real-time analytics of massive IoT data and open research issues. *IEEE Communications Surveys & Tutorials, 19(3)*, 1457–1477.

vom Brocke, J., Hevner, A.R., & Maedche, A. (2020). Introduction to Design Science Research. *Springer Nature Link*. https://doi.org/10.1007/978-3-030-46781-4_1

Walterbusch, M., Grove, S., Breitschwerdt, R., Teuteberg, F., & Thomas, O. (2013). Case-based selection of business process modeling tools: An evaluation criteria framework. *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime, 1*, 686–695.

Weingram, A., Cui, C., Lin, S., Munoz, S., Jacob, T., Viers, J., & Lu, X. (2025). A definition and taxonomy of digital twins: Case studies with machine learning and scientific applications. *Front. High Perform. Comput., Sec. Big Data and AI Volume 3*. https://doi.org/10.3389/fhpcp.2025.1536501

West, S., Stoll, O., Meierhofer, J., & Züst, S. (2021). Digital twin providing new opportunities for value co-creation through supporting decision-making. *Applied Sciences (Switzerland), 11(9), 3750.*

Wieringa, R.J. (2014). Design Science Methodology for Information Systems and Software Engineering. *Springer Berlin*. https://doi.org/10.1007/978-3-662-43839-8

Wu, Q., Di, S., Hou, B., & Bao, Y. (2022). Construction of digital twin model of other-excited DC motor and application of simulation experiments. *9th International Forum on Electrical Engineering and Automation (IFEEA)*. https://doi.org/10.1109/IFEEA57288.2022.10038273

Wu, Y., Zhang, J., Li, Q., & Tan, H. (2023). Research on Real-Time Robust Optimization of Perishable Supply-Chain Systems Based on Digital Twins. *Sensors 2023*. https://doi.org/10.3390/s23041850

Yan, J., Li, X., & Ji, S. (2024). Design and Implementation of Workshop Virtual Simulation Experiment Platform Based on Digital Twin. *Systems 2024, 12, 66.* https://doi.org/10.3390/systems12030066

Zhai, Z. & Chen, H. (2010). Research on business process management framework based on BPEL. *Proceedings - 2010 International Forum on Information Technology and Applications, IFITA 2010, 3*, 45–48.

zur Muehlen, Michael & Rosemann, Michael. (2000). Workflow-based process monitoring and controlling—Technical and organizational issues. *Proceedings of the Hawaii International Conference on System Sciences*, 146.

23 Prilozi

23.1 Prilog 1. Kod neuronskih mreža u Pythonu za održavanje 3D printera

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
import tensorflow as tf
from tensorflow.keras import layers, models
from flask import Flask, jsonify
import random

app = Flask(__name__)

# Global variables to store loaded data and models
test_data = None
trained_models = None

def load_and_preprocess_data(file_path):
    # Load the data
    df = pd.read_csv(file_path)

    # Convert Product ID to categorical features
    df['Product_Type'] = df['Product ID'].str[0]

    # Drop columns we won't use
    df = df.drop(['UID', 'Product ID'], axis=1)

    # Separate numerical and categorical columns
    numerical_columns = ['Air temperature [K]', 'Process temperature [K]',
                         'Torque [Nm]', 'Tool wear [min]']
    categorical_columns = ['Product_Type']

    # Encode categorical variables
    le_type = LabelEncoder()
    le_failure_type = LabelEncoder()

    # Transform categorical columns
    df['Product_Type'] = le_type.fit_transform(df['Product_Type'])
    df['Failure_Type_Encoded'] = le_failure_type.fit_transform(df['Failure Type'])

    # Scale numerical features
    scaler = StandardScaler()
    df[numerical_columns] = scaler.fit_transform(df[numerical_columns])
```

```
    # Prepare features and targets
    X = df[numerical_columns + categorical_columns]
    y_binary = df['Target']
    y_multiclass = df['Failure_Type_Encoded']

    # Split the data
    X_train, X_test, y_binary_train, y_binary_test, y_multi_train, y_multi_test =
train_test_split(
        X, y_binary, y_multiclass, test_size=0.2, random_state=42
    )

    return (X_train, X_test, y_binary_train, y_binary_test, y_multi_train,
y_multi_test,
            le_failure_type.classes_, X.columns, scaler, le_type,
numerical_columns, categorical_columns)


def create_binary_model(input_shape):
    """Create model for binary classification (Failure/No Failure)"""
    model = models.Sequential([
        layers.Dense(64, activation='relu', input_shape=(input_shape,)),
        layers.Dropout(0.2),
        layers.Dense(32, activation='relu'),
        layers.Dropout(0.2),
        layers.Dense(16, activation='relu'),
        layers.Dense(1, activation='sigmoid')
    ])

    model.compile(optimizer='adam',
                  loss='binary_crossentropy',
                  metrics=['accuracy'])
    return model


def create_multiclass_model(input_shape, num_classes):
    """Create model for multiclass classification (Failure Types)"""
    model = models.Sequential([
        layers.Dense(64, activation='relu', input_shape=(input_shape,)),
        layers.Dropout(0.2),
        layers.Dense(32, activation='relu'),
        layers.Dropout(0.2),
        layers.Dense(16, activation='relu'),
        layers.Dense(num_classes, activation='softmax')
    ])

    model.compile(optimizer='adam',
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
    return model
```

```
def train_and_evaluate_models(file_path):
    # Load and preprocess data
    (X_train, X_test, y_binary_train, y_binary_test,
     y_multi_train, y_multi_test, failure_types, feature_names,
     scaler, le_type, numerical_columns, categorical_columns) =
load_and_preprocess_data(file_path)

    # Create and train binary model
    binary_model = create_binary_model(X_train.shape[1])
    binary_history = binary_model.fit(
        X_train, y_binary_train,
        epochs=50,
        batch_size=32,
        validation_split=0.2,
        verbose=1
    )

    # Create and train multiclass model
    multiclass_model = create_multiclass_model(X_train.shape[1],
len(failure_types))
    multi_history = multiclass_model.fit(
        X_train, y_multi_train,
        epochs=50,
        batch_size=32,
        validation_split=0.2,
        verbose=1
    )

    # Evaluate models
    binary_loss, binary_accuracy = binary_model.evaluate(X_test, y_binary_test)
    multi_loss, multi_accuracy = multiclass_model.evaluate(X_test, y_multi_test)

    print("\nBinary Classification (Failure/No Failure):")
    print(f"Test accuracy: {binary_accuracy:.4f}")

    print("\nMulticlass Classification (Failure Types):")
    print(f"Test accuracy: {multi_accuracy:.4f}")

    # Return models and preprocessing info for future predictions
    return {
        'binary_model': binary_model,
        'multiclass_model': multiclass_model,
        'failure_types': failure_types,
        'feature_names': feature_names,
        'scaler': scaler,
        'label_encoder': le_type,
        'numerical_columns': numerical_columns,
        'categorical_columns': categorical_columns
    }
```

```
def make_prediction(models, data_row):
    """Make prediction on a single data row"""
    # Copy the data to avoid modifying the original
    data = data_row.copy().to_frame().T

    # Handle Product ID if present
    if 'Product ID' in data.columns:
        data['Product_Type'] = data['Product ID'].str[0]
        data = data.drop('Product ID', axis=1)

    # Drop UID if present
    if 'UID' in data.columns:
        data = data.drop('UID', axis=1)

    # Encode Product_Type
    if 'Product_Type' in data.columns:
        data['Product_Type'] =
models['label_encoder'].transform(data['Product_Type'])

    # Scale numerical features
    data[models['numerical_columns']] =
models['scaler'].transform(data[models['numerical_columns']])

    # Ensure correct column order
    data = data[models['feature_names']]

    # Make predictions
    failure_prob = models['binary_model'].predict(data)
    failure_type_pred = models['multiclass_model'].predict(data)
    failure_type_indices = np.argmax(failure_type_pred, axis=1)
    failure_types = models['failure_types'][failure_type_indices]

    return {
        'failure_probability': float(failure_prob[0][0]),
        'failure_type': str(failure_types[0])
    }


def load_test_data(test_data_path):
    """Load test data from file"""
    try:
        return pd.read_csv(test_data_path)
    except Exception as e:
        print(f"Error loading test data: {str(e)}")
        return None


@app.route('/printer_status', methods=['GET'])
def get_printer_status():
    global test_data, trained_models
```

```
    if test_data is None or trained_models is None:
        return jsonify({"error": "Test data or models not loaded"}), 500

    # Randomly select a row from test data
    random_index = random.randint(0, len(test_data) - 1)
    selected_row = test_data.iloc[random_index]

    # Get original data for response
    original_data = {
        'product_id': selected_row['Product ID'],
        'air_temperature': float(selected_row['Air temperature [K]']),
        'process_temperature': float(selected_row['Process temperature [K]']),
        'torque': float(selected_row['Torque [Nm]']),
        'tool_wear': int(selected_row['Tool wear [min]'])
    }

    # Make prediction
    prediction = make_prediction(trained_models, selected_row)

    # Prepare response
    response = {
        'timestamp': pd.Timestamp.now().isoformat(),
        'printer_data': original_data,
        'prediction': {
            'failure_probability': prediction['failure_probability'],
            'failure_type': prediction['failure_type'],
           'status': 'WARNING' if prediction['failure_probability'] > 0.5 else 'OK'
        }
    }

    return jsonify(response)


def initialize():
    global test_data, trained_models

    # Train models
    training_file_path = "C:/Doktorski/BIEL/maintenance_training_data.csv"
    trained_models = train_and_evaluate_models(training_file_path)

    # Load test data
    test_file_path = "C:/Doktorski/BIEL/maintenance_test_data.csv"
    test_data = load_test_data(test_file_path)

    if test_data is None:
        print("Failed to load test data. API will not function correctly.")
    else:
        print(f"Successfully loaded {len(test_data)} test data entries.")
```

```
if __name__ == "__main__":
    print("Initializing models and loading test data...")
    initialize()
    print("Starting API server...")
    app.run(debug=True, host='0.0.0.0', port=5000, use_reloader=False)
```

23.2 Prilog 2: Python skripta za pripremu baze podataka u Accessu za proizvodnju dronova

```
import os
import pandas as pd
import pyodbc
import datetime

def set_working_directory(path):
    """Set and confirm the working directory."""
    os.chdir(path)
    print(f"Current working directory: {os.getcwd()}")

def connect_to_database(db_path):
    """Create a connection to the Access database."""
    conn_str = (
        r"Driver={Microsoft Access Driver (*.mdb, *.accdb)};"
        rf"DBQ={db_path};"
    )
    return pyodbc.connect(conn_str)

def drop_table(cursor, table_name):
    """Drop a table if it exists."""
    drop_query = f"DROP TABLE {table_name}"
    try:
        cursor.execute(drop_query)
        cursor.connection.commit()
        print(f"Table '{table_name}' dropped successfully.")
    except pyodbc.Error as e:
        if "does not exist" in str(e).lower():
            print(f"Table '{table_name}' does not exist. Proceeding with
creation.")
        else:
            raise

def create_table(cursor, table_name, schema):
    """Create a table with the given schema."""
    create_query = f"CREATE TABLE {table_name} ({schema})"
    cursor.execute(create_query)
    cursor.connection.commit()
    print(f"Table '{table_name}' created successfully.")
```

```
# Set the working directory
working_directory = "C:\\doktorski\\BIEL\\"
set_working_directory(working_directory)

# Database and table details
access_db = working_directory + "biel.accdb"
tables = {
    "tblJobs": """
        Id Long PRIMARY KEY,
        Status TEXT(10),
        Created DATETIME,
        Finished DATETIME

    """,
    "tblJobStatus": """
        Id AUTOINCREMENT PRIMARY KEY,
        Job_Id LONG,
        Step TEXT(20),
        Status TEXT(10),
        Created DATETIME
    """
}

# Connect to the database
conn = connect_to_database(access_db)
cursor = conn.cursor()

# Drop and recreate tables
for table_name, schema in tables.items():
    drop_table(cursor, table_name)
    create_table(cursor, table_name, schema)

# Close the connection
cursor.close()
conn.close()

print("The database has been prepared.")
```

23.3 Prilog 3: Python skripta za simulator proizvodnje dronova

```
import asyncio
from fastapi import FastAPI
import random
import httpx
from datetime import datetime
import pyodbc
```

```
app = FastAPI()

# Configuration
DB_PATH = "C:\\doktorski\\BIEL\\biel.accdb"
PROCESS_STEPS = ["Warm Up", "Print"]
RESOURCE_INTERVALS = {
    "Warm Up": (1199, 1201),
    "Print": (5399, 5401),
}
JOB_ID_COUNTER = 0

RESOURCE_SEMAPHORES = {
    "Warm Up": asyncio.Semaphore(1),
    "Print": asyncio.Semaphore(1),
}

notification_queue = asyncio.Queue()

async def get_db_connection():
    """Create a connection to the Access database."""
    conn_str = f"Driver={{Microsoft Access Driver (*.mdb,
*.accdb)}};DBQ={DB_PATH};"
    return pyodbc.connect(conn_str)

async def register_job_start(job_id: int):
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO tblJobs (Id, Status, Created) VALUES (?, ?, ?)",
            (job_id, "PENDING", datetime.now())
        )
        conn.commit()
        cursor.close()
        conn.close()
        return {"message": "Job created successfully", "job_id": job_id}
    except Exception as e:
        return {"error": str(e)}

async def register_job_end(job_id: int):
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute(
            "UPDATE tblJobs SET Status = ?, Finished = ? WHERE Id = ?",
            ("COMPLETED", datetime.now(), job_id)
        )
        conn.commit()
        cursor.close()
```

```
        conn.close()
        return {"message": "Job updated successfully", "job_id": job_id}
    except Exception as e:
        return {"error": str(e)}

async def register_job_status(job_id: int, resource_id: str, status: str):
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO tblJobStatus (Job_Id, Step, Status, Created) VALUES (?, ?,
?, ?)",
            (job_id, resource_id, status, datetime.now())
        )
        conn.commit()
        cursor.close()
        conn.close()
        return {"message": "Job status updated successfully", "job_id": job_id}
    except Exception as e:
        return {"error": str(e)}

async def notify_camunda_worker():
    while True:
        message_name, job_id, use_correlation = await notification_queue.get()
        notification_queue.task_done()

async def process_resource_step(step: str, job_id: int):

    semaphore = RESOURCE_SEMAPHORES.get(step)
    if semaphore:
        print(f"[{datetime.now()}] Job {job_id} Waiting For {step}")
        async with semaphore:
            print(f"[{datetime.now()}] Job {job_id} Start {step}")
            await register_job_status(job_id, step, "Start")
            await notification_queue.put((f"{step} Start", job_id, True))
            await asyncio.sleep(random.uniform(*RESOURCE_INTERVALS[step]))
            await register_job_status(job_id, step, "End")
            await notification_queue.put((f"{step} End", job_id, True))
            print(f"[{datetime.now()}] Job {job_id} End {step}")

async def process_job(job_id: int):
    await notification_queue.put(("Job Start", job_id, False))
    await register_job_start(job_id)
    await register_job_status(job_id, "Job", "Start")
    await asyncio.sleep(3)
    for step in PROCESS_STEPS:
       await process_resource_step(step, job_id)
       await asyncio.sleep(3)
    print(f"[{datetime.now()}] Job {job_id} COMPLETED")
    await register_job_status(job_id, "Job", "End")
```

```
    await register_job_end(job_id)

@app.post("/start_job")
async def start_job():
    global JOB_ID_COUNTER
    JOB_ID_COUNTER += 1
    job_id = JOB_ID_COUNTER
    asyncio.create_task(process_job(job_id))
    return {"message": "Job started", "job_id": job_id}
@app.on_event("startup")
async def startup_event():
    """Initialize JOB_ID_COUNTER and start the background task."""
    global JOB_ID_COUNTER
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute("SELECT MAX(Id) FROM tblJobs")
        max_job_id = cursor.fetchone()[0]
        JOB_ID_COUNTER = max_job_id if max_job_id is not None else 0
        cursor.close()
        conn.close()
        print(f"JOB_ID_COUNTER initialized to {JOB_ID_COUNTER}")
    except Exception as e:
        print(f"Error initializing JOB_ID_COUNTER: {e}")

    asyncio.create_task(notify_camunda_worker())

if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

23.4 Prilog 4: Python skripta za *listener* izvođenja procesa proizvodnje dronova

```
import asyncio
from fastapi import FastAPI
import httpx
from datetime import datetime
import pyodbc
from typing import Set, Dict

app = FastAPI()

# Configuration
DB_PATH = "C:\\doktorski\\BIEL\\biel.accdb"
CAMUNDA_URL = "http://localhost:8080/engine-rest/message"
POLL_INTERVAL = 1  # seconds
```

```
# Track processed status entries
processed_status_ids: Set[int] = set()
notification_queue = asyncio.Queue()

async def get_db_connection():
    """Create a connection to the Access database."""
    conn_str = f"Driver={{Microsoft Access Driver (*.mdb,
*.accdb)}};DBQ={DB_PATH};"
    return pyodbc.connect(conn_str)

async def notify_camunda(message_name: str, job_id: int):
    print(f"[{datetime.now()}] Job {job_id} : {message_name}")
    url = CAMUNDA_URL
    payload = {"messageName": message_name}
    if message_name == "Job Start":
        """Send message to Camunda with process variables (initialization)."""
        payload["processVariables"] = {"jobId": {"value": job_id, "type":
"Integer"}}
    else:
        """Send message to Camunda with correlation key."""
        payload["correlationKeys"] = {"jobId": {"value": job_id, "type":
"Integer"}}
    headers = {"Content-Type": "application/json"}

    async with httpx.AsyncClient() as client:
        try:
            response = await client.post(url, json=payload, headers=headers)
            response.raise_for_status()
        except Exception as e:
            print(f"Error sending to Camunda: {e}")

async def notify_camunda_worker():
    """Background worker to process notification queue."""
    while True:
        message_name, job_id = await notification_queue.get()
        await notify_camunda(message_name, job_id)
        notification_queue.task_done()

async def get_new_status_entries():
    """Query new entries from tblJobStatus."""
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute("""
            SELECT Id, Job_Id, Step, Status, Created
            FROM tblJobStatus
            ORDER BY Created ASC
        """)
        entries = [
```

```
            {
                'id': row[0],
                'job_id': row[1],
                'step': row[2],
                'status': row[3],
                'created': row[4]
            }
            for row in cursor.fetchall()
        ]
        cursor.close()
        conn.close()
        return entries
    except Exception as e:
        print(f"Error getting status entries: {e}")
        return []

async def status_monitor():
    """Monitor tblJobStatus and send appropriate messages to Camunda."""
    while True:
        entries = await get_new_status_entries()

        for entry in entries:
            if entry['id'] not in processed_status_ids:
                # Construct message name based on resource and status
                message_name = f"{entry['step']} {entry['status']}"

                if message_name == "Job End":
                    print(f"[{datetime.now()}] Job {entry['job_id']} :
{message_name}")
                else:
                    # Queue notification
                    await notification_queue.put((message_name, entry['job_id']))

                # Mark as processed
                processed_status_ids.add(entry['id'])

        await asyncio.sleep(POLL_INTERVAL)

@app.on_event("startup")
async def startup_event():
    """Start the background tasks."""
    # Start background tasks
    asyncio.create_task(notify_camunda_worker())
    asyncio.create_task(status_monitor())

if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8001)
```

23.5 Prilog 5: Python skripta za pripremu Access baze podataka za proizvodnju mobilnih uređaja

```
import os
import pandas as pd
import pyodbc
import datetime

def set_working_directory(path):
    """Set and confirm the working directory."""
    os.chdir(path)
    print(f"Current working directory: {os.getcwd()}")

def connect_to_database(db_path):
    """Create a connection to the Access database."""
    conn_str = (
        r"Driver={Microsoft Access Driver (*.mdb, *.accdb)};"
        rf"DBQ={db_path};"
    )
    return pyodbc.connect(conn_str)

def drop_table(cursor, table_name):
    """Drop a table if it exists."""
    drop_query = f"DROP TABLE {table_name}"
    try:
        cursor.execute(drop_query)
        cursor.connection.commit()
        print(f"Table '{table_name}' dropped successfully.")
    except pyodbc.Error as e:
        if "does not exist" in str(e).lower():
            print(f"Table '{table_name}' does not exist. Proceeding with
creation.")
        else:
            raise

def create_table(cursor, table_name, schema):
    """Create a table with the given schema."""
    create_query = f"CREATE TABLE {table_name} ({schema})"
    cursor.execute(create_query)
    cursor.connection.commit()
    print(f"Table '{table_name}' created successfully.")

def insert_data(cursor, table_name, columns, data):
    """Insert data into the specified table."""
    placeholders = ", ".join(["?"] * len(columns))
    columns_str = ", ".join([f"[{col}]" for col in columns])
    insert_query = f"INSERT INTO {table_name} ({columns_str}) VALUES
({placeholders})"
```

```
    for _, row in data.iterrows():
        cursor.execute(
            insert_query,
            *[row[col] for col in columns]
        )
    cursor.connection.commit()
    print(f"Data inserted into '{table_name}' successfully.")

# Set the working directory
working_directory = "C:\\doktorski\\MES\\"
set_working_directory(working_directory)

# Load the CSV files
csv_files = {
    "tblLearningData": 'predictive_maintenance.csv',
    "tblResources": 'resources.csv'
}

# Column mappings for each table
column_mappings = {
    "tblLearningData": {
        "Air temperature [K]": "Air_Temperature_K",
        "Process temperature [K]": "Process_Temperature_K",
        "Rotational speed [rpm]": "Rotational_Speed_rpm",
        "Torque [Nm]": "Torque_Nm",
        "Tool wear [min]": "Tool_Wear_Min",
        "Target": "Failure",
        "Failure Type": "Failure_Type"
    },
    "tblResources": {
        "Resource ID": "Id",
        "Description": "Description",
        "Busy": "Busy"
    }
}

# Database and table details
access_db = working_directory + "mes.accdb"
tables = {
    "tblLearningData": """
        Id AUTOINCREMENT,
        Air_Temperature_K SINGLE,
        Process_Temperature_K SINGLE,
        Rotational_Speed_rpm SHORT,
        Torque_Nm SINGLE,
        Tool_Wear_Min SHORT,
        Failure YESNO,
        Failure_Type TEXT(50)
    """,
    "tblOrders": """
```

```
        Id Long PRIMARY KEY,
        Status TEXT(10),
        Created DATETIME,
        Finished DATETIME

    """,
    "tblResources": """
        Id TEXT(20) PRIMARY KEY,
        Description TEXT(50),
        Busy YESNO
    """,
    "tblOrderStatus": """
        Id AUTOINCREMENT PRIMARY KEY,
        Order_Id LONG,
        Resource_Id TEXT(20),
        Status TEXT(10),
        Created DATETIME
    """
}

# Connect to the database
conn = connect_to_database(access_db)
cursor = conn.cursor()

# Drop and recreate tables
for table_name, schema in tables.items():
    drop_table(cursor, table_name)
    create_table(cursor, table_name, schema)

# Load and insert data into tables
for table_name, csv_file in csv_files.items():
    data = pd.read_csv(csv_file)
    column_mapping = column_mappings[table_name]
    data = data[list(column_mapping.keys())]
    data.rename(columns=column_mapping, inplace=True)

    if table_name == "tblLearningData":
        data["Failure"] = data["Failure"].astype(bool).apply(lambda x: -1 if x else
0)  # Convert boolean to Access-compatible

    if table_name == "tblResources":
        data["Busy"] = data["Busy"].astype(bool).apply(lambda x: -1 if x else 0)  #
Convert boolean to Access-compatible

    insert_data(cursor, table_name, list(column_mapping.values()), data)

# Close the connection
cursor.close()
conn.close()
```

```
print("The database has been prepared and the data has been successfully
imported.")
```

23.6 Prilog 6: Kod neuronskih mreža u Pythonu za održavanje bušilice i preše

```
import pyodbc
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.utils import to_categorical
from flask import Flask, request, jsonify

# Database configuration
db_path = "C:\\doktorski\\MES\\mes.accdb"  # DB path
table_name = "tblLearningData"  # table name

# Load data from Access DB
def load_data():
    conn_str = (
        r"Driver={Microsoft Access Driver (*.mdb, *.accdb)};"
        rf"DBQ={db_path};"
    )
    conn = pyodbc.connect(conn_str)
    query = f"SELECT
Air_Temperature_K,Process_Temperature_K,Rotational_Speed_rpm,Torque_Nm,Tool_Wear_Mi
n,Failure,Failure_Type FROM {table_name}"
    data = pd.read_sql(query, conn)
    conn.close()
    return data

# Preprocess the data
def preprocess_data(data):
    # Encode categorical column 'Failure_Type'
    le = LabelEncoder()
    data["Failure_Type"] = le.fit_transform(data["Failure_Type"])

    # Separate features and targets
    X = data.drop(columns=["Failure", "Failure_Type"])
    y_failure = data["Failure"]
    y_failure_type = data["Failure_Type"]

    # Normalize features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
```

```
    # Convert targets to categorical for multi-output neural network
    y_failure_categorical = to_categorical(y_failure)
    y_failure_type_categorical = to_categorical(y_failure_type)

    return X_scaled, y_failure_categorical, y_failure_type_categorical, scaler, le

# Define and train the neural network
def train_neural_network(X, y_failure, y_failure_type):
    input_dim = X.shape[1]
    failure_output_dim = y_failure.shape[1]
    failure_type_output_dim = y_failure_type.shape[1]

    # Define the model
    inputs = Input(shape=(input_dim,))
    x = Dense(64, activation='relu')(inputs)
    x = Dense(32, activation='relu')(x)

    # Two output layers
    failure_output = Dense(failure_output_dim, activation='softmax',
name='failure_output')(x)
    failure_type_output = Dense(failure_type_output_dim, activation='softmax',
name='failure_type_output')(x)

    model = Model(inputs=inputs, outputs=[failure_output, failure_type_output])

    # Compile the model
    model.compile(
        optimizer='adam',
        loss={
            'failure_output': 'categorical_crossentropy',
            'failure_type_output': 'categorical_crossentropy'
        },
        metrics={
            'failure_output': ['accuracy'],
            'failure_type_output': ['accuracy']
        }
    )

    # Train the model
    model.fit(
        X,
        {'failure_output': y_failure, 'failure_type_output': y_failure_type},
        epochs=50,
        batch_size=32,
        verbose=1
    )
    return model

# Load and preprocess the data
```

```
data = load_data()
X, y_failure, y_failure_type, scaler, label_encoder = preprocess_data(data)

# Train the model
model = train_neural_network(X, y_failure, y_failure_type)

# Save the trained model
model.save("predictive_maintenance_model.h5")
print("Model trained and saved successfully.")

# Create a Flask app to serve predictions
app = Flask(__name__)

@app.route('/drill_status', methods=['GET'])
def drillStatus():
    return predict()

@app.route('/press_status', methods=['GET'])
def pressStatus():
    return predict()
def predict():
    try:
        # Connect to the database
        conn_str = (
            r"Driver={Microsoft Access Driver (*.mdb, *.accdb)};"
            rf"DBQ={db_path};"
        )
        conn = pyodbc.connect(conn_str)

        total_records = 10000;

        # Generate a random id
        random_id = np.random.randint(0, total_records)

        # Fetch a random record using the random offset
        query = f"SELECT * FROM {table_name} WHERE Id = {random_id}"
        data = pd.read_sql(query, conn)
        conn.close()

        # Extract and prepare the input data
        sensor_data_array = np.array([[
            data["Air_Temperature_K"].values[0],
            data["Process_Temperature_K"].values[0],
            data["Rotational_Speed_rpm"].values[0],
            data["Torque_Nm"].values[0],
            data["Tool_Wear_Min"].values[0]
        ]])

         # Convert to DataFrame with feature names
```

```
        feature_names = ["Air_Temperature_K", "Process_Temperature_K",
"Rotational_Speed_rpm", "Torque_Nm", "Tool_Wear_Min"]
        sensor_data_df = pd.DataFrame(sensor_data_array, columns=feature_names)

        # Scale input data
        sensor_data_scaled = scaler.transform(sensor_data_df)

        # Predict
        predictions = model.predict(sensor_data_scaled)
        failure_prob = float(predictions[0][0][1])  # Convert to native float
        failure_type_prob = predictions[1][0].tolist()  # Failure type
probabilities
        failure_type =
label_encoder.inverse_transform([np.argmax(failure_type_prob)])[0]

        # Response - Return only failure probability and failure type
        return jsonify({
            "failure_probability": round(failure_prob * 100, 2),
            "failure_type": failure_type
        })
    except Exception as e:
        return jsonify({"error": str(e)})

# Run the Flask app
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

23.7 Prilog 7: Python skripta za MES simulator

```
import asyncio
from fastapi import FastAPI
import random
import httpx
from datetime import datetime
import pyodbc

app = FastAPI()
# Configuration
DB_PATH = "C:\\doktorski\\MES\\mes.accdb"
PROCESS_STEPS = ["Magazine", "Transit to Drill", "Drill", "Transit To Robotino",
"Robotino", "Robotino Return", "Transit To Press", "Press", "Transit to
Collection"]
RESOURCE_INTERVALS = {
    "Magazine": (2, 3),
    "Drill": (5, 10),
    "Robotino": (5, 10),
    "Robotino Return": (5, 10),
    "Press": (5, 10)
}
```

```
ORDER_ID_COUNTER = 0

RESOURCE_SEMAPHORES = {
    "Magazine": asyncio.Semaphore(1),
    "Drill": asyncio.Semaphore(1),
    "Robotino": asyncio.Semaphore(1),
    "Press": asyncio.Semaphore(1)
}

# Dictionary to store events for manual task completion
manual_task_events = {}

notification_queue = asyncio.Queue()

async def get_db_connection():
    """Create a connection to the Access database."""
    conn_str = f"Driver={{Microsoft Access Driver (*.mdb,
*.accdb)}};DBQ={DB_PATH};"
    return pyodbc.connect(conn_str)

async def register_order_start(order_id: int):
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO tblOrders (Id, Status, Created) VALUES (?, ?, ?)",
            (order_id, "PENDING", datetime.now())
        )
        conn.commit()
        cursor.close()
        conn.close()
        return {"message": "Order created successfully", "order_id": order_id}
    except Exception as e:
        return {"error": str(e)}

async def register_order_end(order_id: int):
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute(
            "UPDATE tblOrders SET Status = ?, Finished = ? WHERE Id = ?",
            ("COMPLETED", datetime.now(), order_id)
        )
        conn.commit()
        cursor.close()
        conn.close()
        return {"message": "Order updated successfully", "order_id": order_id}
    except Exception as e:
        return {"error": str(e)}
```

```
async def register_order_status(order_id: int, resource_id: str, status: str):
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO tblOrderStatus (Order_Id, Resource_Id, Status, Created)
VALUES (?, ?, ?, ?)",
            (order_id, resource_id, status, datetime.now())
        )
        conn.commit()
        cursor.close()
        conn.close()
        return {"message": "Order status updated successfully", "order_id":
order_id}
    except Exception as e:
        return {"error": str(e)}

async def notify_camunda_worker():
    while True:
        message_name, order_id, use_correlation = await notification_queue.get()
        notification_queue.task_done()

async def process_robotino_steps(order_id: int):
    """Special handling for Robotino steps that require manual task
confirmation."""
    semaphore = RESOURCE_SEMAPHORES["Robotino"]
    async with semaphore:
        # First Robotino phase
        print(f"[{datetime.now()}] Order {order_id} Start Robotino")
        await register_order_status(order_id, "Robotino", "Start")
        await notification_queue.put(("Robotino Start", order_id, True))
        await asyncio.sleep(random.uniform(*RESOURCE_INTERVALS["Robotino"]))
        await register_order_status(order_id, "Robotino", "End")
        await notification_queue.put(("Robotino End", order_id, True))
        print(f"[{datetime.now()}] Order {order_id} End Robotino")

        # Wait for manual task completion
        print(f"[{datetime.now()}] Order {order_id} Waiting for manual task
completion")
        event = asyncio.Event()
        manual_task_events[order_id] = event
        await event.wait()
        del manual_task_events[order_id]

        # Robotino Return phase
        print(f"[{datetime.now()}] Order {order_id} Start Robotino Return")
        await register_order_status(order_id, "Robotino Return", "Start")
        await notification_queue.put(("Robotino Return Start", order_id, True))
        await asyncio.sleep(random.uniform(*RESOURCE_INTERVALS["Robotino Return"]))
        await register_order_status(order_id, "Robotino Return", "End")
```

```
        await notification_queue.put(("Robotino Return End", order_id, True))
        print(f"[{datetime.now()}] Order {order_id} End Robotino Return")

async def process_resource_step(step: str, order_id: int):
    if step in ["Robotino", "Robotino Return"]:
        if step == "Robotino":  # Only process on Robotino, skip Robotino Return as
it's handled together
            await process_robotino_steps(order_id)
    else:
        semaphore = RESOURCE_SEMAPHORES.get(step)
        if semaphore:
            print(f"[{datetime.now()}] Order {order_id} Waiting For {step}")
            async with semaphore:
                print(f"[{datetime.now()}] Order {order_id} Start {step}")
                await register_order_status(order_id, step, "Start")
                await notification_queue.put((f"{step} Start", order_id, True))
                await asyncio.sleep(random.uniform(*RESOURCE_INTERVALS[step]))
                await register_order_status(order_id, step, "End")
                await notification_queue.put((f"{step} End", order_id, True))
                print(f"[{datetime.now()}] Order {order_id} End {step}")

async def process_order(order_id: int):
    await notification_queue.put(("Order Start", order_id, False))
    await register_order_start(order_id)
    await register_order_status(order_id, "Order", "Start")
    await asyncio.sleep(3)
    for step in PROCESS_STEPS:
        if step.startswith("Transit"):
            print(f"[{datetime.now()}] Order {order_id} In {step}")
            await asyncio.sleep(3)
        else:
            await process_resource_step(step, order_id)
    await notification_queue.put(("Ready To Collect", order_id, True))
    print(f"[{datetime.now()}] Order {order_id} COMPLETED")
    await register_order_status(order_id, "Order", "End")
    await register_order_end(order_id)

@app.post("/start_order")
async def start_order():
    global ORDER_ID_COUNTER
    ORDER_ID_COUNTER += 1
    order_id = ORDER_ID_COUNTER
    asyncio.create_task(process_order(order_id))
    return {"message": "Order started", "order_id": order_id}

@app.post("/manual_task_finished")
async def manual_task_finished(orderId: int):
    print(f"[{datetime.now()}] Manual task finished for Order {orderId}")
    event = manual_task_events.get(orderId)
    if event:
```

```
        event.set()
    return {"message": "Manual task completion received", "order_id": orderId}

@app.on_event("startup")
async def startup_event():
    """Initialize ORDER_ID_COUNTER and start the background task."""
    global ORDER_ID_COUNTER
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute("SELECT MAX(Id) FROM tblOrders")
        max_order_id = cursor.fetchone()[0]
        ORDER_ID_COUNTER = max_order_id if max_order_id is not None else 0
        cursor.close()
        conn.close()
        print(f"ORDER_ID_COUNTER initialized to {ORDER_ID_COUNTER}")
    except Exception as e:
        print(f"Error initializing ORDER_ID_COUNTER: {e}")

    asyncio.create_task(notify_camunda_worker())

if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

23.8 Prilog 8: Python skripta za MES *listener*

```
import asyncio
from fastapi import FastAPI
import httpx
from datetime import datetime
import pyodbc
from typing import Set, Dict

app = FastAPI()

# Configuration
DB_PATH = "C:\\doktorski\\MES\\mes.accdb"
CAMUNDA_URL = "http://localhost:8080/engine-rest/message"
POLL_INTERVAL = 1  # seconds

# Track processed status entries
processed_status_ids: Set[int] = set()
notification_queue = asyncio.Queue()

async def get_db_connection():
    """Create a connection to the Access database."""
```

```
    conn_str = f"Driver={{Microsoft Access Driver (*.mdb,
*.accdb)}};DBQ={DB_PATH};"
    return pyodbc.connect(conn_str)

async def notify_camunda(message_name: str, order_id: int):
    print(f"[{datetime.now()}] Order {order_id} : {message_name}")
    url = CAMUNDA_URL
    payload = {"messageName": message_name}
    if message_name == "Order Start":
        """Send message to Camunda with process variables (initialization)."""
        payload["processVariables"] = {"orderId": {"value": order_id, "type":
"Integer"}}
    else:
        """Send message to Camunda with correlation key."""
        if message_name == "Order End":
            payload = {"messageName": "Ready To Collect"}
        payload["correlationKeys"] = {"orderId": {"value": order_id, "type":
"Integer"}}
    headers = {"Content-Type": "application/json"}

    async with httpx.AsyncClient() as client:
        try:
            response = await client.post(url, json=payload, headers=headers)
            response.raise_for_status()
        except Exception as e:
            print(f"Error sending to Camunda: {e}")

async def notify_camunda_worker():
    """Background worker to process notification queue."""
    while True:
        message_name, order_id = await notification_queue.get()
        await notify_camunda(message_name, order_id)
        notification_queue.task_done()

async def get_new_status_entries():
    """Query new entries from tblOrderStatus."""
    try:
        conn = await get_db_connection()
        cursor = conn.cursor()
        cursor.execute("""
            SELECT Id, Order_Id, Resource_Id, Status, Created
            FROM tblOrderStatus
            ORDER BY Created ASC
        """)
        entries = [
            {
                'id': row[0],
                'order_id': row[1],
                'resource_id': row[2],
                'status': row[3],
```

```
                'created': row[4]
            }
            for row in cursor.fetchall()
        ]
        cursor.close()
        conn.close()
        return entries
    except Exception as e:
        print(f"Error getting status entries: {e}")
        return []

async def status_monitor():
    """Monitor tblOrderStatus and send appropriate messages to Camunda."""
    while True:
        entries = await get_new_status_entries()
                for entry in entries:
            if entry['id'] not in processed_status_ids:
                # Construct message name based on resource and status
                message_name = f"{entry['resource_id']} {entry['status']}"

                # Queue notification
                await notification_queue.put((message_name, entry['order_id']))

                # Mark as processed
                processed_status_ids.add(entry['id'])

        await asyncio.sleep(POLL_INTERVAL)

@app.on_event("startup")
async def startup_event():
    """Start the background tasks."""
    # Start background tasks
    asyncio.create_task(notify_camunda_worker())
    asyncio.create_task(status_monitor())
if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8001)
```

23.9 Prilog 9: Python skripta za vanjsku aktivnost (eng., *worker*)

```
import pycamunda.externaltask

class ExternalTaskException(Exception):
    def __init__(self, *args, message, details='', retry_timeout=10000, **kwargs):
        super().__init__(*args, **kwargs)
        self.message = message
        self.details = details
```

```
        self.retry_timeout = retry_timeout


class Worker:

    def __init__(self, url, worker_id, max_tasks=1, async_response_timeout=5000):
        self.fetch_and_lock = pycamunda.externaltask.FetchAndLock(
            url, worker_id, max_tasks,
async_response_timeout=async_response_timeout
        )
        self.complete_task = pycamunda.externaltask.Complete(
            url, id_=None, worker_id=worker_id
        )
        self.handle_failure = pycamunda.externaltask.HandleFailure(
            url,
            id_=None,
            worker_id=worker_id,
            error_message='',
            error_details='',
            retries=0,
            retry_timeout=0
        )

        self.stopped = False
        self.topic_funcs = {}

    def subscribe(self, topic, func, lock_duration=10000, variables=None):
        self.fetch_and_lock.add_topic(topic, lock_duration, variables)
        self.topic_funcs[topic] = func

    def run(self):
        while not self.stopped:
            tasks = self.fetch_and_lock()

            for task in tasks:
                try:
                    return_variables =
self.topic_funcs[task.topic_name](**task.variables)
                except ExternalTaskException as exc:
                    self.handle_failure.id_ = task.id_
                    self.handle_failure.error_message = exc.message
                    self.handle_failure.error_details = exc.details
                    self.handle_failure.retry_timeout = exc.retry_timeout
                    if task.retries is None:
                        self.handle_failure.retries = 3
                    else:
                        self.handle_failure.retries = task.retries - 1
                    self.handle_failure()
                else:
                    self.complete_task.variables = {}
```

```
                    self.complete_task.id_ = task.id_
                    for variable, value in return_variables.items():
                        self.complete_task.add_variable(name=variable,
value=value)
                    self.complete_task()


def process_collection():
    message = "Collection executed"
    print(message);
    return {'response': message}

def process_deduction(collection_response):
    message = f"Deduction executed based on {collection_response.value}"
    print(message);
    return {'response': message}

def process_action(deduction_response):
    message = f"Action executed based on {deduction_response.value}"
    print(message);
    return {'response': message}

if __name__ == '__main__':
    url = 'http://localhost:8080/engine-rest'
    worker_id = '1'

    worker = Worker(url=url, worker_id=worker_id)
    worker.subscribe(
        topic='collection',
        func=process_collection,
        variables=[]
    )
    worker.subscribe(
        topic='deduction',
        func=process_deduction,
        variables=['collection_response']
    )
    worker.subscribe(
        topic='action',
        func=process_action,
        variables=['deduction_response']
    )
    worker.run()
```

24 Životopis

Snježana Križanić, rođena Svatoš, rođena je 16.12.1989. u Pakracu. Podrijetlom je iz Daruvara. Godine 2008. upisuje Fakultet organizacije i informatike (FOI) u Varaždinu, preddiplomski studij, a godine 2011. upisuje i diplomski studij na istom fakultetu, smjer Organizacija poslovnih sustava. Studij završava 2013. godine s pohvalom *summa cum laude* i težinskim prosjekom ocjena 4.816. U listopadu 2013. se zapošljava u poduzeću Abit d.o.o. na poziciji Java programera gdje obavlja poslove razvoja i održavanja bankarskih aplikacija. U lipnju 2018. se zapošljava na poziciji projektne asistentice na FOI-ju na projektu „*Razvoj inovativne platforme za digitalnu transformaciju poduzeća*" te počinje održavati nastavu iz laboratorijskih vježbi iz kolegija: Modeliranje poslovnih procesa, Poslovni procesi u organizaciji. U veljači 2019. upisuje Doktorski studij smjer Informacijske znanosti na FOI-ju. U ožujku 2020. se zapošljava na poziciji asistentice u nastavi na FOI-ju te održava nastavu iz laboratorijskih vježbi iz kolegija: Baze podataka 1, Baze podataka 2. Trenutno održava nastavu iz laboratorijskih vježbi iz kolegija: Procesno servisne arhitekture, Digitalno poslovanje i Elektroničko poslovanje na Katedri za razvoj informacijskih sustava na Fakultetu organizacije i informatike. Autorica je niza radova iz područja digitalne transformacije, modeliranja poslovnih procesa, procesnog rudarenja i digitalnih blizanaca. Od stranih jezika služi se vrlo dobro engleskim, dobro njemačkim i talijanskim te češkim jezikom, koji joj je materinji. Područja od interesa su joj digitalni blizanci, poslovni procesi, izrada procesno orijentiranih aplikacija i izrada pametnih ugovora u Solidity programskom jeziku. Uz nabrojano, Snježana Križanić je ponosna majka dviju djevojčica, Lune i Anabele.

25 Popis objavljenih znanstvenih radova

Prilog u časopisu
Izvorni znanstveni rad

1. **Križanić, Snježana**; Vrček, Neven
Development of a Process Digital Twin in Camunda Modeler // TEM Journal, 13 (2024), 4; 3315-3323. doi: 10.18421/TEM134-66

2. **Križanić, Snježana**
Educational data mining using cluster analysis and decision tree technique: A case study // International journal of engineering business management, 12 (2020), 1-9. doi: 10.1177/1847979020908675

3. Hrustek, Larisa ; Kutnjak, Ana ; **Križanić, Snježana**
Promjene u marketinškim procesima pod utjecajem digitalne transformacije s fokusom na korisničko iskustvo // CroDiM, 3 (2020), 1; 98-106

4. Pihir, Igor ; **Križanić, Snježana** ; Kutnjak, Ana
Digitalna transformacija marketinga u malim i srednjim poduzećima - pregled postojećih istraživanja // CroDiM, 2 (2019), 1; 125-134

Prilog sa skupa (u zborniku)
Izvorni znanstveni rad

1. Kaniški, Matija; **Križanić, Snježana**
State-of-the-Art Machine Learning Frameworks for Training or Inference on Business Process Dataset // Proceedings of 47th ICT and Electronics Convention (MIPRO) / Skala, Karolj; Mornar, Vedran (ur.).
Rijeka: Croatian Society for Information, Communication and Electronic Technology - MIPRO, 2024. str. 1865-1870. doi: 10.1109/MIPRO60963.2024.10569816

2. **Križanić, Snježana**; Vrček, Neven; Habuš, Matija
From Classroom to Real-World: University Students' Experience in Learning Solidity and Blockchain // 35th International Scientific Conference CECIIS 2024 / Korent, Dina; Vrček, Neven (ur.).
Varaždin: University of Zagreb, Faculty of Organization and Informatics, 2024. str. 91-97

3. Rabuzin, Kornelije ; Cerjan, Maja ; **Križanić, Snježana**
Supporting Data Types in Neo4j // ADBIS 2022. Communications in Computer and Information Science, vol 1652..
Cham: Springer, 2022. str. 459-466. doi: 10.1007/978-3-031-15743-1_42

4. Rabuzin, K. ; Modrušan, N. ; **Križanić, S**. ; Kelemen, R.
Process Mining in Public Procurement in Croatia // Proceedings on 18th International Conference on Industrial Systems – IS'20.
Springer, 2022. str. 473-480. doi: 10.1007/978-3-030-97947-8_62

5. **Križanić, Snježana** ; Rabuzin, Kornelije
Process Mining and Data Warehousing – A Literature Review // 31st Central European Conference on Information and Intelligent Systems : CECIIS 2020 : Proceedings / Strahonja, Vjeran; Steingartner, William; Kirinić, Valentina (ur.).
Varaždin: Faculty of Organization and Informatics, University of Zagreb, 2020. str. 33-39

6. **Križanić, Snježana** ; Šestanj-Perić, Tanja ; Kutnjak, Ana
ERP Solutions in Cloud Technologies as a Driver for Digital Transformation of Businesses // Proceedings of the 43rd International Convention on Information, Communication and Electronic Technology 2020.
Rijeka: Hrvatska udruga za informacijsku i komunikacijsku tehnologiju, elektroniku i mikroelektroniku - MIPRO, 2020. str. 1538-1543

7. Kutnjak, Ana ; Hrustek, Larisa ; **Križanić, Snježana**
Applying the Decision Tree Method in Identifying Key Indicators of the Digital Economy and Society Index (DESI) // Proceedings of the 43rd International Convention on Information, Communication and Electronic Technology 2020.
Rijeka: Hrvatska udruga za informacijsku i komunikacijsku tehnologiju, elektroniku i mikroelektroniku - MIPRO, 2020. str. 1576-1581

8. Kutnjak, Ana ; **Križanić, Snježana** ; Pihir, Igor
Educational and practical view of knowledge, skills and experience needed by a chief digital officer // EDULEARN19 Proceedings.
Palma de Mallorca: International Academy of Technology, Education and Development (IATED), 2019. str. 5711-5718. doi: 10.21125/edulearn.2019.1389

9. Hrustek, Larisa ; Kutnjak, Ana ; **Križanić, Snježana**
Promjene u marketinškim procesima pod utjecajem digitalne transformacije s fokusom na korisničko iskustvo // 4thInternational Scientific and Professional Conference(CRODMA 2019): Book of Papers, Theme:marketing 4.0.
Zagreb: Hrvatska udruga za direktni i interaktivni marketing (CRODMA), 2019. str. 92-100

10. **Križanić, Snježana** ; Šestanj Perić, Tanja ; Tomičić- Pupek, Katarina ;
The changing role of ERP and CRM in Digital Transformation // Economic and Social Development, 41st International Scientific Conference on Economic and Social Development, Book of Proceedings.
Beograd: VADEA ; Megatrend univerzitet ; Sveučilište Sjever ; Faculty of Management University of Warsaw ; Faculty of Law, Economics and Social Sciences Sale Mohammed V University in Rabat, 2019. str. 248-256

11. **Križanić, Snježana** ; Hrustek, Larisa ; Tomičić- Pupek, Katarina
Raising the readiness for using digital technologies in teaching processes // EDULEARN19 Proceedings.
Palma de Mallorca: International Academy of Technology, Education and Development (IATED), 2019. str. 5601-5607. doi: 10.21125/edulearn.2019.1369

12. **Križanić, Snježana** ; Tomičić-Pupek, Katarina
Process parameters discovery based on application of k-means algorithm - a real case experimental study // Proceedings of 30th International Scientific Conference : Central European Conference on Information and Intelligent Systems 2019 / Strahonja, Vjeran; Hertweck, Dieter; Kirinić, Valentina (ur.).
Varaždin: Faculty of Organization and Informatics, 2019. str. 223-229

Pregledni rad (znanstveni)

1. Habuš, Matija; Vrček, Neven; **Križanić, Snježana**
Optimization of urban road network and digital twins: a literature review // 35th International Scientific Conference CECIIS 2024 / Korent, Dina; Vrček, Neven (ur.).
Varaždin: University of Zagreb, Faculty of Organization and Informatics, 2024. str. 99-105

2. **Križanić, Snježana**; Vrček, Neven
Business process modeling and digital twins: a literature review // 34th Central European Conference on Information and Intelligent Systems: Proceedings / Vrček, Neven; Marcos Ortega, Luis de; Grd, Petra (ur.).
Zagreb: Sveučilište u Zagrebu, Fakultet organizacije i informatike, 2023. str. 145-151