



NAMJENSKI JEZIK I OKRUŽENJE ZA MODELOVANJE GREŠAKA U PROIZVODNIM PROCESIMA I GENERISANJE DOKUMENTACIJE FMEA

DOMAIN SPECIFIC LANGUAGE AND TOOL FOR MODELING ERRORS IN PRODUCTION PROCESSES AND GENERATION OF FMEA DOCUMENTATION

Đordije Ivanišević, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – PRIMJENJENE RAČUNARSKE NAUKE I INFORMATIKA**

**Kratak sadržaj** – U industriji se trenutno događa četvrta revolucija čiji cilj je da upotrebom savremenih tehnologija odgovori na zahtjeve tržišta za brzom adaptacijom proizvoda, a posljedično i procesa proizvodnje. U situaciji kada se proizvodni procesi brzo mijenjaju, mogu se pojaviti greške koje je teško predvidjeti što povećava cijenu troškova proizvodnje. U ovom radu predstavljeni su namjenski jezik i okruženje za modelovanje grešaka u proizvodnim procesima, koji bi mogli uticati na povećanje stepena automatizacije i smanjenje cijene troškova proizvodnje. Na osnovu modela proizvodnih procesa koji su rezultat upotrebe predloženog namjenskog jezika moguće je automatski generisati dokumentaciju FMEA i time skratiti vrijeme potrebno za ručno pisanje dokumentacije i smanjiti broj grešaka koji se u takvom procesu javlja.

**Ključne reči:** Namjenski jezici, namjensko modelovanje, razvoj vođen modelima, proizvodni procesi

**Abstract** – The fourth revolution is currently taking place in the industry, with the goal to respond to the market demands for the rapid product and process adaptation. When production processes are changing rapidly, errors often occur and are difficult to predict and process. Thus, the manufacturing costs are increased. In this paper, a domain-specific language and a tool for the production process and error modeling are presented. Their purpose is to increase the degree of automation and to reduce the manufacturing costs. Based on the production process models, it is possible to automatically generate FMEA documentation and thus shorten the time required to write the documentation manually which could reduce the number of errors that occur in such process.

**Keywords:** Domain-specific languages, domain-specific modeling, model-driven development, production processes

**1. UVOD**

U industriji se trenutno događa četvrta revolucija koja je nastala iz potrebe za brzom adaptacijom proizvodnje kako bi bilo moguće proizvoditi visoko prilagođene proizvode za korisnike. U tradicionalnom pristupu proizvodnji, svaka izmjena na ciljnom proizvodu zahtjevala bi rekonfiguraciju proizvodnog pogona što je vremenski

zahtjevno i utiče na povećanje troškova proizvodnje. Upotrebom savremenih tehnologija koje omogućavaju integraciju i sinhronizaciju fizičkih i virtualnih aspekata proizvodnih pogona teži se skraćanju vremena potrebnog za rekonfiguraciju. Fabrike koje omogućavaju takav vid integracije nazivaju se pametne fabrike. Pametne fabrike uvijek su „svjesne“ trenutnog stanja, donose odluke u realnom vremenu analizom velike količine podataka i tako omogućavaju svim učesnicima proizvodnog procesa da obavljaju svoje zadatke u što kraćem vremenskom periodu.

U situaciji kada se proizvodni procesi često mijenjaju, teško je isprojektovati proizvodni proces tako da bude tolerantan na greške. U ovom radu predložen je namjenski jezik i okruženje koji bi omogućili modelovanje grešaka u proizvodnim procesima na formalan i precizan način. U tradicionalnom pristupu proizvodnji, odlučivanje o izvršavanju sekvence operacija koje otklanjaju uočenu grešku zasniva se na procedurama koje nisu dovoljno automatizovane. Operater u trenutku pojave greške mora „ručno“ da otkloni grešku kako bi se proizvodnja nastavila. Prednost pristupa sa predloženim namjenskim jezikom ogleđa se u tome što bi izvršavanje koraka koji obrađuju grešku bilo automatizovano na osnovu modela procesa proizvodnje.

Upotrebom predloženog namjenskog jezika, podaci o kontrolnim aktivnostima i potencijalnim greškama u proizvodnji bili bi integrisani u model proizvodnog procesa. Takve podatke bilo bi moguće iskoristiti za automatsko generisanje dokumentacije, što bi proces kreiranja dokumentacije značajno ubrzalo ali i smanjilo broj grešaka koje su posledica ručnog kreiranja dokumentacije. Ciljna dokumentacija za generisanje predložena u ovom radu preporučena je metodologijom FMEA (engl. *Failure Modes and Effect Analysis*). Metodologija FMEA koristi istorijat uočenih grešaka za analizu, ublažavanje efekata i prevenciju pojave grešaka u budućnosti. Glavni cilj metodologije FMEA jeste stvaranje uslova za konfiguraciju proizvodnog pogona u kome se posljedice greške uočavaju i eliminišu prije nego što proizvod dođe do krajnjeg korisnika [1].

Korišćenjem namjenskog jezika za modelovanje grešaka u proizvodnim procesima bilo bi moguće povećati automatizaciju direktnim izvršavanjem specificiranih koraka od strane mašina i smanjiti troškove proizvodnje. Takođe, postojanje formalnog jezika omogućilo bi lakše i fleksibilnije modelovanje izmjena proizvodnog procesa i grešaka koje bi se mogle javiti kao posljedica tih izmjena.

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Vladimir Dimitrieski, docent.

## 2. PREGLED POSTOJEĆEG STANJA U OBLASTI

### 2.1. Postojeći načini za modelovanje proizvodnih procesa

Potreba za modelovanjem i opisivanjem proizvodnih procesa razvila se i prije pojave računara. Neke fabrike i danas koriste razne vidove nedigitalne tehničke dokumentacije koja opisuje tokove materijala, redoslijed izvršavanja operacija, učesnike procesa, kontrolu kvaliteta i slično. Međutim, da bi bilo moguće vršiti validaciju modela proizvodnih procesa i transformacije u druge modele ili tekst, neophodno je da model bude kreiran upotrebom formalnog jezika sa jasno definisanom sintaksom. Neki od jezika koji se koriste za modelovanje proizvodnih procesa su: Petrijeve mreže, BPMN (engl. *Business Process Model and Notation*) i UML (engl. *Unified Modeling Language*).

Modelovanje kompleksnih proizvodnih procesa upotrebom UML-a i Petrijevih mreža rezultuje modelima koji nisu dovoljno čitljivi i pregledni. U poređenju sa UML-om i Petrijevim mrežama, BPMN sadrži koncepte koji su više prilagođeni iskazivanju strukture procesa. Međutim, koncepti BPMN-a orijentisani su ka modelovanju poslovnih procesa i nisu dovoljno prilagođeni domenu proizvodnje. Kreiranjem proširenja za prethodno nabrojane jezike mogli bi se riješiti prethodno nabrojani nedostaci. Međutim, proširenja često povećavaju kompleksnost modela i projektanti proizvodnih procesa nisu navikli na njihovu sintaksu.

Razvoj namjenskog jezika za modelovanje proizvodnih procesa koji je, sa jedne strane razumljiv krajnjem korisniku, a sa druge strane dovoljno specijalizovan da može opisati sve koncepte iz domena, trebalo bi da riješi prethodno nabrojane probleme.

### 2.2. Namjenski jezici

Jezici za modelovanje služe da za iskazivanje informacija ili znanja o posmatranom sistemu, pri čemu imaju tačno definisan skup sintaksnih pravila. Prilikom modelovanja projektanti moraju poštovati sintakсна pravila formalno definisanih jezika za modelovanje. Postoje dvije osnovne kategorije u koje se mogu svrstati jezici za modelovanje: namjenski jezici i jezici opšte namjene. Jezici opšte namjene mogu se koristiti za opis sistema nezavisno od konkretnog domena. Namjenski jezici predstavljaju jezike specijalno kreirane za određeni kontekst kako bi olakšali opis problematike usko vezane za posmatrani domen. Kako modelovanje proizvodnih procesa zahtjeva rad sa konceptima usko vezanim za domen proizvodnje, u ovom radu predložen je pristup modelovanja procesa upotrebom namjenskog jezika.

Prilikom razvoja namjenskog jezika neophodno je definisati njegovu apstraktnu i konkretnu sintaksu. Apstraktnu sintaksu definišu meta-modeli. Meta-modeli sadrže definiciju jezika za modelovanje i omogućavaju kreiranje klasa modela reprezentovanih tim jezikom [2]. Vizualna notacija konceptata uvedenih u meta-modelu definisana je konkretnom sintaksom.

U zavisnosti od načina iscrtavanja elemenata modela postoje dvije glavne kategorije konkretnih sintaksi: tekstualna i grafička. Upotrebom tekstualnih konkretnih sintaksi informacije o modelu se kodiraju sekvencama karaktera, dok grafička sintaksa podrazumijeva kodiranje

informacija o modelu u vidu prostorno uređenih grafičkih i tekstualnih simbola.

Za implementaciju namjenskog jezika za modelovanje grešaka u proizvodnim procesima izabrana je grafička konkretna sintaksa. Glavni razlog za to je da bude omogućena lakša vizualizacija grešaka za korisnike koji nemaju programerskog iskustva tj. nisu navikli na tekstualne programske jezike.

### 2.3. Modelom vođeni razvoj

Modelom vođene (engl. *Model-Driven*) metodologije se mogu definisati kao metodologije koje koriste prednosti modela u inženjerskim aktivnostima. Prednosti korišćenja modela su: mogućnost istraživanja različitih rješenja u ranoj fazi razvoja, kreiranje više različitih pogleda na sistem, lakše otkrivanje grešaka i brži razvoj proizvoda. Osim toga, korišćenjem modela se povećava nivo apstrakcije za posmatrani sistem i tako se omogućava bolje razumijevanje kompleksnih sistema.

U modelom vođene paradigme spada i modelom vođeni razvoj softvera, odnosno MDSD (engl. *Model Driven Software Development*). Primjenom metodologije MDSD modeli se koriste u svim fazama životnog ciklusa jednog softverskog proizvoda. Dakle, primjenjuju se u svrhe opisivanja korisničkih zahtjeva, kreiranja statičkih analiza, kreiranja apstraktne predstave sistema, generisanja dokumentacije i kreiranja automatskih testova.

Definisanjem mapiranja između različitih modela moguće je vršiti transformacije modela koje takođe predstavljaju bitan aspekt paradigme MDSD. Transformacije se definišu na nivou meta-modela, a primjenjuju na nivou modela. Transformacije se takođe mogu posmatrati kao modeli, odnosno kao instance meta-modela koji ih opisuje. Prilikom razvoja predloženog namjenskog jezika korišćene su transformacije modela procesa u dokument FMEA.

## 3. IMPLEMENTACIJA NAMJENSKOG JEZIKA ZA MODELOVANJE GREŠAKA U PROIZVODNIM PROCESIMA

### 3.1. Apstraktna sintaksa

Meta-model namjenskog jezika za upravljanje greškama definiše njegovu apstraktnu sintaksu. Za razvoj meta-modela korišćen je EMF (engl. *Eclipse Modeling Framework*). EMF je radni okvir koji se koristi za kreiranje meta-model upotrebom jezika *Ecore* i za generisanje alata zasnovanog na meta-modelu. Kako sva ograničenja nije moguće definisati upotrebom jezika *Ecore*, za specifikaciju složenijih ograničenja korišćen je OCL (engl. *Object Constraint Language*). Na slici 1. prikazan je meta-model jezika za modelovanje grešaka u proizvodnim procesima.

*Model* je korijenski koncept koji u sebi sadrži sve ostale koncepte. Unutar *Model*-a se definišu proizvodni procesi koji su modelovani sa *Manufacture* konceptom. Globalni obrađivač grešaka (*GlobalErrorHandler*) predstavlja zaseban proces koji služi za korigovanje grešaka. Svrha ovog koncepta je da pojednostavi modelovanje proizvodnog procesa u situaciji kada postoji više različitih grešaka koje se obrađuju na isti način. Dakle, i

*Manufacture* i *GlobalErrorHandler* su procesi i zbog toga nasljeđuju apstraktnu klasu *Process*. Oba koncepta mogu da sadrže sljedeće komponente: procesne elemente (*ProcessElement*), veze (*Relationship*) i kontrolne aktivnosti (*Inspection*). *ProcessElement* je predstavljen apstraktnom klasom i njegova svrha jeste generalizacija dva elementa proizvodnog procesa – procesnih koraka i kapija.

Procesni korak (*ProcessStep*) služi za modelovanje tačno jednog zadatka u proizvodnom procesu, a kapija (*Gate*) za modelovanje kontrole toka izvršavanja. Kapije mogu biti tipa: **PARALLELISM** – omogućavaju kreiranje paralelnih tokova i **DECISION** – omogućavaju kreiranje odluke u kontroli toka. *Relationship* se koristi za modelovanje veza između procesnih elementa, a *Inspection* za modelovanje kontrolnih aktivnosti koje otkrivaju greške.

### 3.2. Konkretna sintaksa

Za razvoj grafičke konkretne sintakse korišćen je *Sirius*. *Sirius* je okruženje koje služi za kreiranje grafičke sintakse jezika i omogućava automatsko generisanje alata za modelovanje.

Za prikaz podataka iz modela proizvodnog procesa korišćene su tri različite dijagramske reprezentacije: *ModelRepresentation*, *ManufactureRepresentation* i *GlobalHandlerRepresentation*. *ModelRepresentation* omogućuje prikaz modela sa modelovanim proizvodnim procesima prikazanim u vidu grafičkih simbola. *ManufactureRepresentation* prikazuje unutrašnju strukturu svakog od modela proizvodnih procesa, dok *GlobalHandlerRepresentation* prikazuje modele procesa koji opisuju obradu grešaka nastalih u proizvodnim procesima. Sve tri reprezentacije imaju definisana mapiranja grafičkih simbola za elemente modela (tabela 1.), kao i alate za manipulaciju nad tim elementima. *ManufactureRepresentation* je jedina reprezentacija koja ima dva dodatna sloja: *Manufacture* i *QualityAssurance*. Dijagrami kompleksnih proizvodnih procesa obuhvataju previše informacija što ih čini nepreglednim, te je svrha organizovanja komponenti kroz slojeve da omogući svakom korisniku da vidi samo one dijelove modela proizvodnog procesa koji su njemu neophodni. *Manufacture* sloj sadrži komponente koje su inženjeru

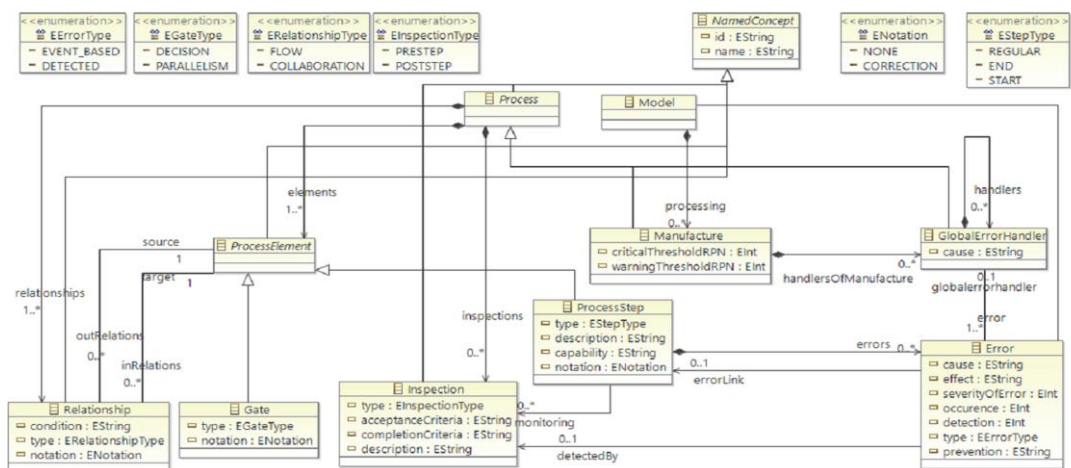
proizvodnih procesa potrebne za modelovanje strukture proizvodnog procesa, dok *QualityAssurance* ima komponente potrebne inženjeru kontrole kvaliteta da modeluje potencijalne greške i kontrolne aktivnosti.

Na slici 2 prikazan je primjer modela proizvodnog procesa za proizvodnju majica. U primjeru su modelovane tri greške uzrokovane događajem i jedna detektovana greška. Detektovana greška se otkriva tokom izvršavanja kontrolne aktivnosti, dok se greška uzrokovana događajem desila u neočekivanim uslovima.

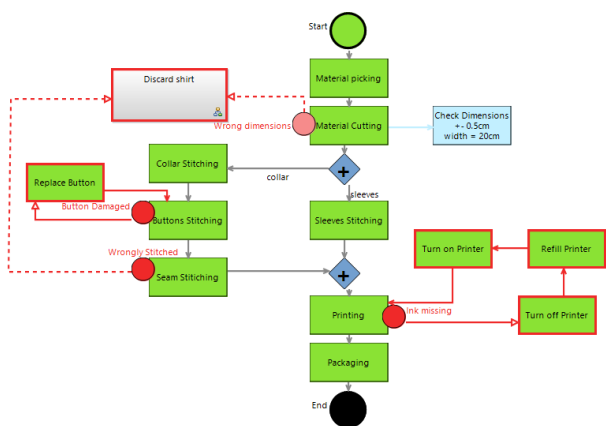
Greške sa nazivima *Wrong dimensions* i *Wrongly stiched* obrađuju se posredstvom globalnog obrađivača greške *Discard shirt*. Greške sa nazivima *Ink missing* i *Button Damged* obrađuju se lokalno – korišćenjem sekvence korektivnih procesnih koraka.

Tabela 1. Grafički simboli svih elemenata modela

Element	Simbol
Regularni procesni korak	
Korektivni procesni korak	
Regularna kapija odlučivanja	
Regularna kapija paralelizma	
Korektivna kapija odlučivanja	
Korektivna kapija paralelizma	
Detektovana greška	
Greška uzrokovana događajem	
Kontrolna aktivnost	
Globalni obrađivač greške	
Regularna veza između procesnih elemenata	
Veza između procesnog koraka i kontrolne aktivnosti	
Veza između greške i korektivnog procesnog koraka	
Veza između greške i globalnog obrađivača greške	
Veza između dva korektivna procesna koraka	



Slika 1. Meta-model namjenskog jezika za modelovanje grešaka u proizvodnim procesima



Slika 2. Primjer modela procesa za proizvodnju majica

#### 4. GENERISANJE DOKUMENTACIJE FMEA

Formalno kreiran model proizvodnog procesa upotrebom predloženog namjenskog jezika, moguće je iskoristiti za automatsko generisanje digitalne dokumentacije preporučene metodologijom FMEA.

Process step	Potential failure mode	Potential effect(s) of error	Severity	Potential cause(s) of error	Occurrence	Current process controls prevention	Current process controls detection	Detection	Current error handler	RPN
Buttons Stitching	Button Damaged	Time consuming	4	Invalid input material	3	Following work instruction	Error occured without control mechanisms	5	LOCAL: Starts with step: Replace Button	60
Seam Stitiching	Wrongly Stitiched	Material Discarded	7	Machine excecuted wrong operation	1	Following work instrunctions	Error occured without control mechanisms	5	GLOBAL: Discard shirt	35
Printing	Ink missing	Time consuming	5	Bad monitoring	2	Following work instrunctions	Error occured without control mechanisms	2	LOCAL: Starts with step: Turn off Printer	20
Material Cutting	Wrong dimensions	Time loss, material loss	7	Operator mistake	8	Following work instrunctions	Check Dimensions : Visual control	9	GLOBAL: Discard shirt	504

Slika 3. Dokumentacija FMEA za proizvodnju majica

Generator koda za HTML tabelu sa sadržajem FMEA dokumenta parsira strukturu modela sve dok se ne dođe do *Error*-a. Vrijednosti svih atributa koncepta *Error* se potom čitaju i koriste za kreiranje sadržaja dokumenta FMEA.

#### 5. ZAKLJUČAK

Namjenski jezik predložen u ovom radu omogućava kreiranje preciznih i formalnih modela grešaka u proizvodnji i procesa koji te greške otklanjaju. Izborom grafičke konkretne sintakse prilikom specifikacije jezika omogućeno je stvaranje okruženja koje je intuitivno krajnjim korisniku. Okruženje predloženo u ovom radu implementirano je tako da ga koriste dva tipa korisnika: inženjeri proizvodnih procesa i inženjeri kontrole kvaliteta. Inženjeri proizvodnih procesa koriste okruženje da kreiraju modele koji opisuju strukturu proizvodnog procesa sa tačno definisanim koracima proizvodnje. Tako definisan model sa jasno utvrđenim granicama procesa, inženjeri kontrole kvaliteta proširuju konceptima koji formalno opisuju kontrolne aktivnosti, pojavu i obradu grešaka. Implementiran je generator koji na osnovu modela proizvodnog procesa generiše dokumentaciju FMEA. Svrha dokumenta FMEA jeste da se u budućnosti izbjegnu pojave uočenih i predviđenih grešaka. To se postiže analizom prethodnih grešaka i uočavanjem kritičnih koraka proizvodnje koja predstavljaju područja visokog rizika u smislu posljedica koja ostavljaju na cijeli sistem.

Neka od mogućih proširenja rada su konkretizacija jezika na određenu granu industrije, lokalizacija okruženja, dodavanje tekstualne konkretne sintakse kao i implementacija generatora za druge tipove

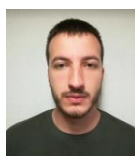
Metodologija FMEA se koristi prilikom analize rizika u oblasti proizvodnih procesa. Rizik greške se posmatra iz tri aspekta: ozbiljnost greške u pogledu njenog uticaja na korisnika proizvoda, broj proizvoda koji imaju neki vid oštećenja na uzorku od jednog miliona proizvoda i vjerovatnoća uočavanja greške. Svaka greška sadrži atribute koji opisuju nabrojane aspekte. Atributima se dodjeljuje cjelobrojna vrijednost iz intervala od 1 do 10. Ukupni rizik greške ocjenjuje se na osnovu broja RPN (engl. *Risk Priority Number*) koji se računa množenjem pomenute tri vrijednosti [3]. Na nivou proizvodnog procesa projektant definiše dva praga vrijednosti za RPN koji svaku grešku klasifikuju u neki od tri opsega za procjenu rizika greške: nizak, srednji ili visok. Vrijednost RPN-a za pomenute opsege u ciljnoj dokumentaciji se redom prikazuje sa zelenom, žutom i narandžastom pozadinom. Na taj način projektantima je skrenuta pažnja na greške sa većim stepenom rizika kako bi izmijenili proizvodni proces u cilju prevencije te greške ili ublažavanja njenog efekta. Izgenerisana dokumentacija za primjer modela sa slike 2. prikazana je na slici 3.

dokumentacija. Konkretizacijom jezika za određenu granu industrije modeli bi bili prilagođeniji logici proizvodnje iz tog sektora. Interfejs okruženja za modelovanje prilagođen je korisnicima sa engleskog govornog područja. Lokalizacija okruženja na druge jezike bi omogućila i korisnicima koji ne razumiju engleski jezik da efektivno kreiraju modele. Dodavanjem tekstualne konkretne sintakse, inženjerima sa programerskim iskustvom bi proces modelovanja bio ubrzan.

#### 6. LITERATURA

- [1] C. P. Ahire, A S. Relklar, "Correlating Failure Modes Effect Anlysis (FMEA) & Overall Equipment Effectiveness (OEE)", *Procedia Engineering* 38, ISSN 3482-3486, 2012, doi: [10.1016/j.proeng.2012.06.402](https://doi.org/10.1016/j.proeng.2012.06.402)
- [2] M. Brambilla, J. Cabo, M. Wimmer, "Model-Driven Software Engineering in Practise"(Second edition), Morgan & Claypool Publishers, 2017
- [3] M. Svingerova, M. Melichar, "Evaluation of Process Risks in Industry 4.0 Environment", *Proceedings of the 28th DAAAM International Symposium*, pp. 1021-1029, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, 2017 doi: [10.2507/28th.daaam.proceedings.142](https://doi.org/10.2507/28th.daaam.proceedings.142)

#### Kratka biografija:



**Đordije Ivanišević** rođen je u Nevesinju 1996. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Primjenjene računarske nauke i informatika odbranio je 2019.god.  
kontakt: [djordij3 ivanis@uns.ac.rs](mailto:djordij3_ivanis@uns.ac.rs)