



UNIVERSITATEA DIN PITESTI
ACADEMIA MAGISTRA VITAE

Școala Doctorală – Informatică

TEZĂ DE DOCTORAT

Tehnici De Testare Și Validare Aplicate În Robotică

REZUMAT

Doctorand:

Cristina Nicoleta Mazilu

Conducător Științific:

Prof. univ. dr. Tudor Bălănescu

Pitești 2021

CUPRINS TEZĂ

Listă figuri

Listă Abrevieri

Capitolul 1. Introducere

- 1.1 Scop și Obiective
- 1.2 Structura Tezei
- 1.3 Contribuția Tezei

Capitolul 2. Concepte fundamentale

- 2.1 Verificarea Modelelor (Model Checking) pentru un mecanism robotizat
 - 2.1.1 Verificarea Modelelor (Model Checking), Event-B, Rodin și ProB
 - 2.1.2 Automate cu stări finite (Finite State Machines - FSM)
 - 2.1.3 Cinematica unui mecanism robotizat
 - 2.1.4 Publicații referitoare la verificarea Modelelor mecanismelor robotizate
- 2.2 Modele de gestionare a cunoștințelor (Knowledge Models) pentru testarea software-ului în robotică
 - 2.2.1 Managementul Cunoștințelor (Knowledge Management)
 - 2.2.2 Knowledge Management în Testare Software folosind Mutanți
 - 2.2.3 Simulare în robotică
 - 2.2.4 Modele de cunoștințe (Knowledge Models) în Celonis Studio
 - 2.2.5 Managementul Cunoștințelor (Knowledge Management) în robotică
- 2.3 Minarea proceselor (Process mining) unui mecanism robotizat
 - 2.3.1 Concepte fundamentale despre minarea proceselor (Process mining)
 - 2.3.2 Fundamente despre BPMN
 - 2.3.3 Publicații relevante referitoare la verificarea conformității procesului în robotică

Capitolul 3. Modelarea și Verificarea Modelelor în Robotică

- 3.1 Metodologie
- 3.2 Un Model FSM (Finite State Machine) pentru un brat robot

3.3 Studiu de Caz: Un manipulator cu două articulații $r-\theta$

3.3.1 Modelul Event-B

3.3.2 Validarea Modelului

Capitolul 4. Modele de Gestionare a Cunoștințelor pentru Testarea Software a Roboților

4.1 Metodologie

4.2 Experimentare

4.2.1 Rezultate în cazul corect pentru KPI – Unghiul generat de un obstacol la dreapta

4.2.2 Rezultate în cazul mutant pentru KPI – Unghiul generat de un obstacol la dreapta

4.2.3 Rezultate în cazul corect pentru KPI – Unghiul generat de un obstacol la stânga

4.2.4. Rezultate în cazul mutant pentru KPI – Unghiul generat de un obstacol la stânga

Capitolul 5. Minarea Procesului unui Mecanism Robotizat

5.1 Metodologie

5.2 Experimentare

5.2.1 Modelul mișcării de rotație de la FSM la BPMN 2.0

5.2.2 Modelul mișcării de translație de la FSM la BPMN 2.0

5.2.3 Verificarea conformității procesului mișcării de rotație

5.2.4 Verificarea conformității procesului mișcării de translație

5.3 Transformarea Statisticilor Procesului în Acțiune

Capitolul 6. Concluzii și Activitatea Viitoare

6.1 Rezumat

6.2. Contribuții

6.3 Avantaje și Limitări

6.4 Activitatea Viitoare

Bibliografie

CUPRINS REZUMAT

Capitolul 1. Introducere	6
1.1 Structura Tezei.....	7
Capitolul 2. Concepte fundamentale	8
2.1 Utilizarea Model Checking pe mecanisme robot.....	8
2.2 Automate cu stări finite	9
2.3 Cinematica unui mecanism robotizat.....	10
2.4 Modele de gestionare a cunoștințelor pentru testarea software-ului în robotică (KM)	10
2.5 Managementul cunoștințelor în testarea software	11
2.6 Simulatoarele în robotică	12
2.7 Modele de cunoștințe în Celonis Studio	13
2.9 Managementul cunoștințelor în industria robotică	13
2.9 Minarea proceselor mecanismelor robotice.....	14
2.10 BPMN (Business Process Modelling and Notation)	15
2.11 Rezultate	16
Capitolul 3. Verificarea Modelelor în robotică	18
3.1 Metodologie	18
3.2 Un Model FSM al unui braț robot.....	19
3.3 Un manipulator cu două articulații - studiu de caz	22
3.4 Modelul Event-B	22
3.5 Verificarea formală a Modelului	22
Capitolul 4. Modele de cunoștințe pentru testarea software în robotică	24
4.1 Metodologie	24
4.2 Experimentare.....	25
Capitolul 5. Minarea proceselor pe un mecanism robotizat	28
5.1 Metodologie	28
5.2 Experimentare.....	29
5.3 Verificarea conformității procesului de rotație.....	31
5.4 Transformarea Statisticilor procesului în acțiune folosind Celonis Action Engine.....	31
Capitolul 6. Concluzii și activitatea viitoare	33
6.1 Contribuții	33
6.2 Avantaje și limitări	34

6.3 Activitatea viitoare.....	35
Bibliografie	37

CUVINTE CHEIE

Model checking, Event-B, Rodin, BPMN (Business Process Model and Notation), FSM (Automate cu stări finite), Managementul Cunoștințelor (Knowledge Management), Testare Software, Webots, Minarea Proceselor (Process Mining), YAML, Celonis.

Capitolul 1. Introducere

Metodologiile combinate propuse în această teză urmăresc să contribuie la asigurarea calității roboților și mecanismelor robotizate de la proiectare până la funcționalitate. Scopul acestei teze este explorarea și descrierea metodelor de testare menite să evalueze funcționalitatea sistemelor robotizate atât în simulare, cât și în realitate.

Robotica este una dintre principalele domenii de cercetare în zilele noastre și au fost dezvoltate mai multe metodologii de testare, totuși utilizarea modelării matematice (FSM) combinată cu verificarea modelelor (Model Checking) este o nouă abordare pentru simularea și testarea comportamentului unui robot în faza de proiectare. O etapă cheie a fazei de dezvoltare este partea de testare a software-ului, care este abordată prin utilizarea unui Sistem de Gestionare a Cunoștințelor (Knowledge Management System), deoarece aceasta susține toate funcțiile necesare unei sistem de asigurare a calității, de la crearea de cunoștințe despre cum ar trebui să funcționeze sistemul în fiecare caz, până la schimbul de cunoștințe și actualizarea între echipele de dezvoltare și testare. Pe lângă luarea în considerare și aplicarea verificării modelului în faza de proiectare și a managementului cunoștințelor în faza de testare software, este descrisă o tehnică de verificare a conformității procesului care utilizează un instrument de minare a proceselor (Process Mining) în faza operațională. În contextul Industriei 4.0, care utilizează sisteme Cloud pentru a permite gestionarea centralizată a datelor, deoarece Minarea Proceselor este o soluție bazată pe Cloud, această tehnică ar putea fi adoptată în Industria 4.0, pentru prevenirea posibilelor incidente în procesul operațional al mecanismelor robotizate.

Pe scurt, obiectivele acestei teze sunt următoarele: (i) utilizarea tehnicii Model checking pentru verificarea anumitor proprietăți ale unui mecanism robotizat. Se sugerează că această metodologie este utilă pentru ca proiectanții roboților să poată detecta și remedia problemele în cazul unei încălcări a specificațiilor. Metodologia este aplicată pe un studiu de caz, o platformă software (Rodin) fiind utilizată ca instrument de modelare și verificare formală. (ii) descrierea și aplicarea unei metodologii de gestionare a cunoștințelor (Knowledge Management) pentru testarea software a unui robot. Metodologia este aplicată unui studiu de caz obținut folosind simulatorul Webots. (iii) sugerarea unei metodologii pentru partea de testare operațională a unui braț robot. Metodologia constă în utilizarea tehnicii de extragere a proceselor (Process Mining) pentru

verificarea conformității unui mecanism robotizat. Platforma Celonis este utilizată în acest caz, datorită facilităților privind automatizarea proceselor în timp real.

1.1 Structura Tezei

Această teză este organizată în șase capitole, primele două fiind dedicate *Introducerii* (Capitolul 1) și *Conceptelor Fundamentale* (Capitolul 2).

Capitolul 3 - *Modelarea si Verificarea Modelelor în Robotică*, prezintă rezultatele originale concentrate pe partea de modelare și verificare a modelelor mecanismelor robotizate în faza de proiectare.

Capitolul 4 - *Modele de Gestionare a Cunoștințelor pentru Testarea Software a Roboților*, prezintă rezultatele originale obținute de autoare pe parcursul procesului de cercetare, concentrat pe partea de utilizare a unui sistem de management al cunoștințelor pentru sisteme robotizate și autonome în timpul fazei de testare.

Capitolul 5 - *Minarea Procesului unui Mecanism Robotizat*, prezintă rezultatele originale obținute de autoare pe parcursul procesului de cercetare, concentrat pe o metodologie legată de aplicarea unei tehnici de extragere a proceselor pentru verificarea conformității unui mecanism robotizat în faza operațională.

Capitolul 6 - *Concluzii și Activitatea Viitoare*, conține o sinteză structurată a principalelor rezultate ale acestei teze în termeni de metodologie, provocări de implementare și rezultate. De asemenea, sunt discutate anumite limitări rezultate din complexitatea industriei roboticii. La final sunt prezentate perspective ale direcțiilor viitoare de cercetare legate de subiectul abordat.

Capitolul 2. Concepte fundamentale

Conceptele utilizate în această teză sunt introduse în secțiunile acestui capitol. Fundamentele utilizate în această teză sunt legate de abordările de verificare a modelelor roboților în faza de prototip, de gestionare a cunoștințelor în faza de testare software și de verificare a conformității proceselor roboților în faza operațională.

2.1 Utilizarea Model Checking pe mecanisme robot

FSM-urile ar putea reprezenta o tehnică adecvată pentru a formaliza Modelarea sistemelor robotizate complexe. Un Model Event-B al fiecărui FSM este implementat în platforma Rodin, validat formal folosind mecanisme încorporate de verificare automată a consistenței capabile să detecteze diferite erori, cum ar fi blocaje sau încălcări ale unor proprietăți LTL corespunzătoare anumitor comportamente ale robotului folosind verificatorul de modele ProB. Aceasta s-a dovedit a fi o metodă eficientă de testare a Modelului robotului.

Modelele Event-B sunt organizate în jurul a două construcții de bază: contexte și mașini. Contextele sunt menite să descrie partea statică a unui Model, iar mașinile specifică partea dinamică. Rolul contextelor este de a separa parametrii unui Model formal și proprietățile corespunzătoare, care se presupune că sunt valabile pentru toate cazurile. O mașină descrie un sistem a cărui stare este dată de un set de variabile și tranzițiile fiind Modelate prin intermediul unui set de evenimente. Mașinile specifică proprietățile comportamentale ale Modelelor Event-B. Pentru a accesa informațiile despre contextul C , mașina M trebuie să fie asociată cu C . Când mașina M vede contextul C , aceasta are acces la mulțimile și constantele lui C , la axiome și teoreme, pentru a le folosi ca ipoteze în timpul verificării (Hoang Thai Son, 2013). Se obține o execuție corectă dacă la fiecare pas, toate obligațiile de validare (proof obligations) sunt adevărate.

Un eveniment este o tranziție între stări structurate după cum urmează:

Event nameOfTheEvent

when

guard1 . . . guardn

then

action1 . . . actionn

end

Gărzile (guard1, ..., guardn) sunt condiții care trebuie îndeplinite pentru a permite activarea unui eveniment. Un eveniment fără nici o garda va fi întotdeauna activat.

Acțiunile (acțiunea1, ..., acțiunea n) reprezintă descrierea modului în care apariția evenimentului contribuie la modificarea unora dintre variabilele mașinii. Toate acțiunile unui eveniment sunt efectuate simultan. O acțiune poate fi fie deterministă (folosind operatorul de atribuire: =), fie nedeterministă (folosind operatorul: $\in$). O acțiune nedeterministă are forma generală $x: \in \{\text{set de valori posibile}\}$ și implică atribuirea unei valori aleatorii din setul de valori posibile variabilei x (Abrial JR, 2007).

2.2 Automate cu stări finite

Potrivit lui Paul E Black (2016): O mașină cu stări finite (FSM) este un model de calcul care este alcătuit dintr-un set de stări: o stare de pornire, un alfabet de intrare, o funcție de tranziție între stări care ia ca intrări un simbol de intrare și o stare actuală și returnează stare următoare.

O gamă variată de FSM-uri au fost definite începând cu automatele Moore (Moore Edward F 1956) și Mealy (Mealy George H, 1955) și mai recent mașinile UML (Douglass Bruce Powel, 1999) și automatele cu caracter ierarhic (Vanderson HF, Adenilso S, Reza MM, 2019).

În mod formal, un FSM determinist este un cvintuplu $(\Sigma, S, s_0, \text{Act}, F)$, unde:

- Σ este un alfabet finit, nevid
- S este o mulțime finită, nevidă de stări
- $s_0 \in S$ este starea inițială
- $\text{Act}: S \times \Sigma \rightarrow S$ este funcția de tranziție a stărilor

- $F \subset S$ este mulțimea stărilor finale

Cuvântul determinist se referă la faptul că ori de câte ori este definit $\text{Act}(s, x)$, din orice stare s și pentru orice intrare x , tranziția este unică.

2.3 Cinematica unui mecanism robotizat

Referindu-se la mișcarea corpurilor fără a lua în considerare forțele care au provocat mișcarea, cinematica este unul dintre cele mai importante aspecte ale proiectării, controlului și simulării robotului (Waldron K, Schmiedeler J, 2008).

Mecanismele robotice sunt sisteme de mai multe corpuri rigide interconectate prin articulații. Astfel, cinematica unui robot descrie aspecte precum: viteza, accelerația sau poziția și orientarea corpurilor în spațiu. Cele mai comune topologii utilizate pentru conectarea corpurilor sunt topologia în serie și mecanismele complet paralele. În timp ce, într-un sistem serializat, fiecare corp, cu excepția primului și ultimului, este conectat la alte două, într-un mecanism complet paralel, două componente sunt interconectate prin articulații multiple (Waldron K, Schmiedeler J, 2008).

2.4 Modele de gestionare a cunoștințelor pentru testarea software-ului în robotică (KM)

Managementul cunoștințelor aplicat în testarea software-ului acceptă toate funcțiile necesare unei activități de Quality Assurance, de la crearea de cunoștințe despre modul în care sistemul ar trebui să funcționeze în fiecare caz, până la partajarea și actualizarea cunoștințelor în cadrul echipei. Cu alte cuvinte, adoptând KM pentru procesele de testare software de bază, se obțin beneficii în termeni de cost, calitate și eficiență (K. Wnuk și T. Garrepalli 2018).

Managementul cunoștințelor a câștigat popularitate în diferite industrii în ultimele două decenii, permițând proceselor să adauge valoare și să fie competitive. Ceea ce înțelegem acum prin Managementul cunoștințelor este informația axată pe rezultate (P. Drucker, 1985), reflectată în

progresele tehnologice și transformată într-o disciplină. În abordarea tradițională, un sistem de gestionare a cunoștințelor este de obicei considerat un container care trebuie umplut cu cunoștințe extrase de la experți, în care tehnologia de Modelare a cunoștințelor permite sistemului de asistare a deciziilor să ajute un operator în detectarea și diagnosticarea problemelor într-un sistem dinamic (Cuenca, J. și Molina, M. 2000).

Dintre aplicațiile KM, cel mai cunoscut termen este Ontologia. Termenul de ontologie provine din filozofie, unde reprezintă o explicație sistematică a lucrurilor. O ontologie înseamnă cunoștințe corelate stocate de exemplu într-un sistem, care poate fi utilizat pentru a răspunde la interogări sau pentru a introduce date către alte sisteme (Noy, N. & McGuinness, Deborah, 2001).

Metodele și instrumentele de inginerie a cunoștințelor oferă abordări disciplinate pentru stocarea și accesarea cunoștințelor, iar un număr semnificativ de companii utilizează această entitate valoroasă pentru a îndeplini sarcini și a crea noi cunoștințe.

Independent de nivel, activitățile efectuate în ingineria cunoștințelor sunt aceleași. Aceste tranziții sunt numite ciclul de evoluție a cunoștințelor. Conform Wiig (1999), acesta este un ciclu cu cinci faze. Ciclul începe cu crearea de cunoștințe în organizație; informațiile despre cunoștințe sunt capturate în forme explicite; cunoștințele explicite sunt structurate și clasificate; iar cunoștințele tacite și explicite sunt accesate și aplicate (Wiig, 1999).

2.5 Managementul cunoștințelor în testarea software

Ingineria software este solicitantă și are nevoie de acces de mare viteză la entitățile de cunoaștere. KM permite inginerilor să acceseze informațiile necesare despre sistemul în curs de dezvoltare și, de asemenea, să creeze noi cunoștințe de expertiză.

Testarea este obligatorie în orice dezvoltare de software, iar testarea bazată pe mutații a fost clasificată drept una dintre cele mai eficiente tehnici de testare. Testarea folosind mutații folosește tehnicile de injectare a erorilor. B. Falah, S. Bouriat, O. Achahbar (2015) explică faptul că dacă mutantul generează un rezultat diferit de cel așteptat, acest lucru sugerează că programul conține o eroare sintactică care trebuie corectată. În acest caz, suita de testare este considerată eficientă, iar mutantul este denumit mutant ucis. În caz contrar, dacă suita de testare nu a reușit să identifice

prezența unei erori, atunci aceasta este inadecvată și se spune că mutantul este în viață (S. Hamimoune, B. Falah, 2016).

E. F. de Souza, R. de Almeida Falbo și colab. (2015) concluzionează că tendința este spre un interes crescând în abordarea Managementului cunoștințelor în testarea software-ului. Majoritatea lucrărilor menționate în studiul lor evidențiază aspecte precum beneficiul suportului automat pentru gestionarea cunoștințelor în testare software prin intermediul unui Sistem de Gestionare a Cunoștințelor.

2.6 Simulatoarele în robotică

Interesul imens pentru tehnicile de Machine Learning a sporit nevoia de definire a sistemelor KM, indiferent de industriile și domeniile în care activează companiile. Când se ia în considerare Industria Roboticii, entitățile de cunoaștere și necesitatea asociată de seturi de date relevante, au dat naștere cererii de utilizare a simulatoarelor roboților. Cu alte cuvinte, platforma de simulare a roboților devine un instrument la îndemână pentru dezvoltarea Sistemelor de Gestionare a Cunoștințelor care pot susține învățarea din comportamentul simulat al robotului și pot fi verificate. Simularea poate fi utilizată pentru a îmbunătăți roboții deja dezvoltați și pentru a testa roboții care nu au fost încă lansați pe piață.

Atunci când se iau în considerare tehnicile de Inteligență Artificială (AI) capabile să îmbunătățească generația de roboți prin adăugarea de abilități decizionale, este clar că validarea acestor roboți înainte de implementare este obligatorie. Folosirea instrumentelor de simulare combinate cu sisteme KM, poate accelera procesul de testare (HeeSun Choi, Cindy Crump și colab., 2021).

Simulatoarele pentru roboți sunt implementate folosind numeroase limbaje de programare precum Python, C, C ++, Java, C # și alte limbaje care decid compatibilitatea platformei lor (K. Kumar și PS Reel, 2011), cu toate acestea, independent de platformă, este esențial să testăm comportamentul robotului luând în considerare fiecare schimbare posibilă a mediului. Simulatoare precum Cyberbotics Webots, Energid's Actin și Laminar Research's X-Plane sunt câteva exemple de instrumente comerciale răspândite. În scopul acestei teze, Webots a fost ales datorită faptului că

este o platformă open-source care permite simulări pentru roboți deja existenți pe piață precum Aibo, Lego mindstorms, Khepera, Koala, Hermission, BoeBot, E-puck, iRobot și altele. (K. Kumar și P. S. Reel, 2011).

2.7 Modele de cunoștințe în Celonis Studio

Celonis este astăzi liderul global în Managementul Execuției Proceselor, Printre partenerii tehnologici ai Celonis se numără companii precum IBM, Amazon Web Services, Microsoft, Oracle și Salesforce. Mai mult decât atât, Celonis oferă parteneriate Academic Alliance pentru mai mult de 400 de instituții academice din întreaga lume, valorificând conexiunile dintre academie și industria Celonis (Celonis, 2021).

În prezent, companiile utilizează entități de cunoștințe precum definiții, înregistrări, indicatori de performanță, variabile, filtre etc., care pot fi create, partajate și întreținute în întreaga organizație utilizând diferite tehnologii și instrumente. Aceste entități de cunoștințe pot fi implementate folosind un limbaj de reprezentare a cunoștințelor sau structuri de date care permit interpretarea cunoștințelor de către oameni și software în același timp. Modelele de cunoștințe în Celonis (Celonis, 2021), pot fi considerate ca un dicționar de entități importante de gestionare a cunoștințelor care permit controlul versiunilor și sunt configurate în limbajul YAML.

2.9 Managementul cunoștințelor în industria robotică

Industria roboticii este un domeniu emergent și complex, de aceea este nevoie de utilizarea unor instrumente avansate pentru a crea, stoca, utiliza, întreține și actualiza cunoștințele necesare dezvoltării și manipulării roboților. Datorită complexității acestui domeniu, există încă multe aspecte legate de Managementul Gestionării Cunoștințelor în robotică care trebuie explorate și îmbunătățite.

O soluție care vizează rezolvarea problemelor de gestionare a cunoștințelor și de comunicare legate de robotică este publicată de T. Tammet, E. Reilent, M. Puju și colab. (2010). Lucrarea descrie implementarea unei arhitecturi de cunoaștere a roboților de curățare iRobot Roomba și soluția propusă este inspirată din principiile web semantice.

Datorită disponibilității companiilor de a face față cerințelor clienților, scop în care trebuie să facă îmbunătățiri continue asupra produselor lor, un studiu al lui Mușat F., Miha F. (2018) explorează utilizarea unui sistem adecvat de gestionare a cunoștințelor. Scopul cercetării este de a utiliza un robot programabil, capabil să îndeplinească sarcini repetitive. Robotul ar trebui să fie capabil să se adapteze la orice proces utilizat într-o companie din industria auto.

Un alt studiu relevant în direcția managementului cunoștințelor bazat pe ontologie aplicat în robotică este prezentat de L. V. Gómez, J. Miura (2021). În acest articol este descrisă dezvoltarea unui sistem cu module specifice capabile să gestioneze cunoștințele și raționamentul roboților, analiza comenzilor, luarea deciziilor și interacțiunea.

În concluzie, este necesară aplicarea managementului cunoștințelor în testarea software în industria robotică, totuși există încă multe modalități neexplorate cu ajutorul cărora această abordare ar putea fi îmbunătățită.

2.9 Minarea proceselor mecanismelor robotice

Minarea proceselor a fost introdusă de Wil Van der Aalst în 2004 și s-a concentrat inițial pe analiza datelor stocate în sistemele IT.

Procesul de minare este o tehnică emergentă care are originea în Business Intelligence și reprezintă o combinație de extragere a datelor, modelare și analiză a proceselor. În prezent, minarea proceselor are ca scop descoperirea, verificarea și îmbunătățirea proceselor în timp real, conectându-se la cunoștințele din jurnalele de evenimente disponibile în sistemele informaționale de astăzi (Wil Van der Aalst, 2004).

Principalele tehnici de extragere a proceselor sunt:

- *Play-in (Discovery)*

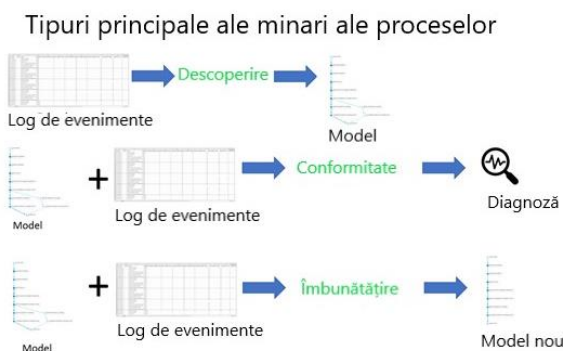
Comportamentele sunt preluate din jurnalele de evenimente pentru a construi un Model (proces).

- *Play-out*

Utilizează un model de proces pentru a genera comportamente (de exemplu, prin simulare) pentru a înțelege procesul.

- *Replay*

Constă în utilizarea unui jurnal de evenimente și a unui model de proces ca intrare. Jurnalul de evenimente este comparat cu modelul standard pentru a analiza diverse fenomene, de exemplu, deviații și predicții. Această tehnică este utilă pentru verificarea și îmbunătățirea proceselor.



Tipuri de minări ale proceselor

2.10 BPMN (Business Process Modelling and Notation)

Din limbajele grafice existente pentru BPM (Business Process Modelling), cea mai robustă comunicare între sistemele IT și operațiunile organizațiilor este realizată folosind tehnica BPMN, În 2004, Object Management Group (OMG) a dezvoltat un model de proces de notație standard (BPMN 2020). După finalizarea primelor revizui majore ale BPMN, OMG a lansat BPMN 2.0 în 2011 (Aagesen Gustav & Krogstie John, 2015).

Potrivit lui Aagesen Gustav & Krogstie John (2015), dezvoltarea procesului (BPMN) este inspirată din mai multe limbaje anterioare de modelare a proceselor care adoptă o abordare transformativă (input-process-output). Procesele sunt împărțite în activități, care pot fi împărțite în continuare în subactivități. Această perspectivă a fost aleasă din standardele unor limbaje transformatorii bine

cunoscute precum IDEF (IDEF-3x 1993) și rețelele Petri (Van der Aalst WMP, Desel J, Oberweis A, 2000).

2.11 Rezultate

Următoarele capitole au la bază publicații ale autoarei acestei teze.

Capitolul 3, are la bază următoarea publicație: *Adrian Turcanu & Talal Shaikh & Cristina Nicoleta Mazilu (Turcanu) (2020). Despre verificarea Modelului unui mecanism robotizat. Journal of Robotics and Automation. Volumul 4 | Numărul 1 | DOI: 10.36959 / 673/366*

<https://scholars.direct/Articles/robotics/jra-4-018.php?jid=robotics>

Această lucrare se concentrează pe utilizarea Event-B și a platformei Rodin pentru verificarea Modelului unui mecanism robotizat. Aceste instrumente au fost alese datorită următoarelor caracteristici: simplitatea Modelării, validarea automată a Modelelor și facilitățile de verificare și animare a Modelelor. Robustețea unui Model Event-B este înrădăcinată în rigoarea limbajului matematic din spate și în corectitudinea Modelului care este asigurată în fiecare etapă a procesului de Modelare prin diferite mecanisme automate. Mai mult, tehnica de rafinare permite utilizatorilor să dezvolte Modele treptat și astfel să includă în Modelul actual doar acele caracteristici esențiale în scopul etapei respective. Deși metodologia propusă și demonstrată are unele restricții care decurg din limitările limbajului Event-B și ale platformei Rodin, acest studiu ar putea reprezenta un pas semnificativ în direcția verificării Modelelor aplicate în industria robotică.

Capitolul 4 este derivat din publicația autoarei: Cristina Nicoleta Turcanu (2021). Modele de cunoștințe pentru testarea software-ului în robotică. A 15-a conferință internațională despre tehnologia informației și aplicații (ICITA) – în curs de publicare în volumul conferinței; programul conferinței: https://www.icita.world/files/ICITA_2021.pdf.

Această lucrare prezintă o metodologie propusă de testare software folosind Modele de cunoștințe, aplicate pentru roboți. Un mediu de simulare pentru roboți a fost folosit, deoarece simulatoarele sunt mai rapide, mai puțin costisitoare și mai convenabile de utilizat. Platforma Webots a fost

utilizată în acest scop pentru a genera amprentele procesului bazate pe mutanți. Pe de altă parte, Modelele de cunoștințe au fost dezvoltate folosind un alt instrument comercial puternic, Celonis EMS, care oferă și o licență academică de software gratuit. Pentru implementarea Modelului de cunoștințe, a fost utilizat limbajul standard de serializare a datelor YAML, fiind deja integrat cu Celonis. Lucrarea este concentrată în mai multe etape: (i) Modificarea codului robotului în Webots pentru a genera jurnale de evenimente ale unor activități simulate (ii) Conectarea jurnalelor de evenimente cu platforma Celonis, (iii) Crearea în Celonis EMS Studio, al Modelului de cunoștințe al robotului (iv) Adăugarea unor definiții KPI în Modelul de cunoștințe v) Utilizarea Modelului de cunoștințe obținut în vederea testării software folosind mutanți. Metodologia a fost demonstrată folosind un studiu de caz al unui robot de aspirare iRobot obținut folosind platforma Webots.

Capitolul 5 este derivat din publicația autoarei: *Cristina Nicoleta Turcanu (2021). Minarea proceselor unui un mecanism robotizat. IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2021, pp. 205-212, doi: 10.1109 / ICSTW52544.2021.00043.*

<https://ieeexplore.ieee.org/document/9440183>

Metodologia propusă în această lucrare are în vedere investigarea conformității procesului pentru un mecanism robotizat. Datorită provocărilor de a colecta un flux de date operaționale industriale reale, pentru simplitate, un proces simulat cu unele caracteristici neconforme care trebuie verificate este investigat folosind platforma Celonis și sunt explorate unele acțiuni automate în caz de neconformitate. Lucrarea se concentrează pe prezentarea unei metodologii împărțite în mai multe faze (conversia FSM la Modelul BPMN pentru un mecanism robotizat, testarea conformității procesului și acțiunile corespunzătoare în caz de neconformitate).

Capitolul 3. Verificarea Modelelor în robotică

În acest capitol este demonstrat modul în care verificarea Modelelor (Model Checking) poate fi utilă în faza de proiectare a unui mecanism robotizat (un braț robotizat). Cinematica unui robot este reprezentată de o succesiune de corpuri legate de articulații. Fiecare mișcare a brațului robotizat poate fi reprezentată ca un flux de rotații și translații și se construiesc FSM-urile corespunzătoare comportamentului fiecăreia dintre acestea. Metodologia dată este aplicată unui studiu de caz simplu: un manipulator cu 2 articulații. Modelele Event-B pentru cele două FSM rezultate sunt implementate în platforma Rodin, iar acestea sunt validate folosind ProB, verificatorul asociat.

3.1 Metodologie

Această secțiune descrie metodologia aplicării verificării Modelelor pentru un robot folosind Event-B. Pașii acestui proces sunt următorii:

1. Citirea specificațiilor mecanismului robotizat
2. Proiectarea FSM-urilor asociate cu cinematica robotului
3. Implementarea în platforma RODIN a Modelului Event-B al fiecărui FSM
4. Validarea în mod formal a fiecărui Model
5. Verificarea proprietăților fiecărui Model specificat folosind LTL
6. Dacă se observă vreo eroare, se revine la Modelul Event-B și se reconsidera proiectarea Modelului Event-B
7. Dacă Modelul Event-B nu este corect, se revine la specificațiile mecanismului robotic, se indentifică sursa erorii și se corectează.

3.2 Un Model FSM al unui braț robot

O mișcare complexă este descrisă ca un produs al mai multor matrice ale unor rotații și translații. Pornind de la această idee, sunt introduse două categorii de FSM: una corespunzătoare unei rotații cu un unghi dat și cealaltă corespunzătoare unei translații cu o lungime dată. Brațul se poate roti cu până la 180 de grade în sensul acelor de ceasornic și invers acelor de ceasornic și poate fi extins până la o lungime maximă fixă, extensia fiind măsurată de la punctul său final.

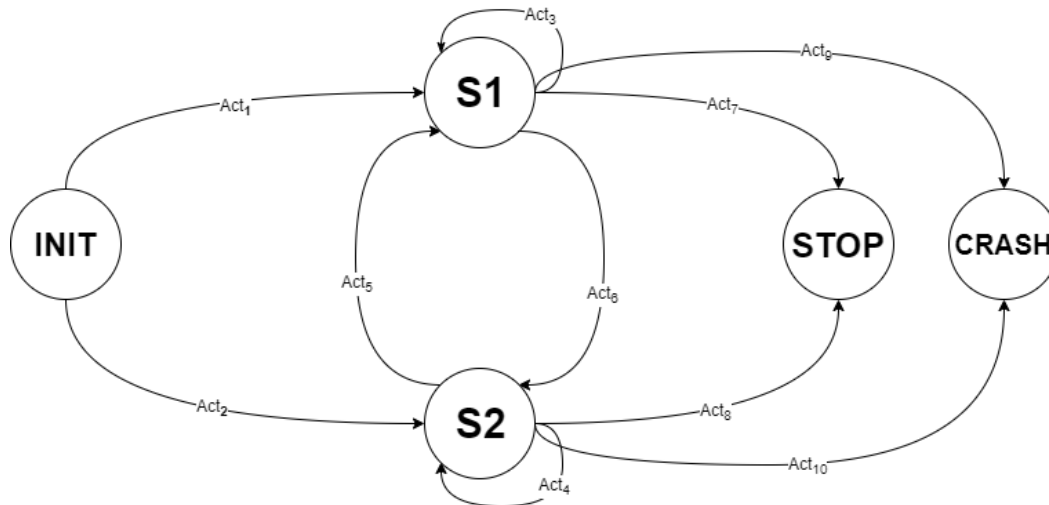
FSM-ul de rotație are următoarele componente:

- alfabetul, $\Sigma = \{\theta\}$
- mulțimea de stări, $S = \{\text{INIT}, s1, s2, \text{STOP}, \text{CRASH}\}$
- starea inițială, $s0 = \text{INIT}$
- mulțimea de acțiuni $\text{ACT} = \{\text{act}_i \mid 1 \leq i \leq 10\}$
- și mulțimea de stări finale $F = \{\text{STOP}, \text{CRASH}\}$

Reprezentarea sa grafică este dată în figura următoare, iar semnificația acțiunilor este următoarea:

- act1 corespunde rotației brațului robot în sens invers acelor de ceasornic cu un unghi θ , unde $0 < \theta < 180$; în acest caz starea se va schimba în s1
- act2 corespunde unei rotații în sensul acelor de ceasornic cu un unghi θ , unde $-180 < \theta < 0$; în consecință, starea se schimbă în s2.
- act3 și act4 corespund rotațiilor cu orice unghi θ astfel încât stările s1 și respectiv s2 să nu fie modificate.
- act5 și act6 corespund rotațiilor cu un unghi astfel încât starea să se schimbe de la s1 la s2 sau invers. Un astfel de scenariu este posibil ori de câte ori semnul unghiului de rotație se schimbă ca urmare a unei rotații.
- act7 corespunde unei tranziții între starea s1 și starea STOP ($\theta = -180$)
- act8 corespunde unei tranziții între starea s2 și starea STOP ($\theta = 180$)

- în mod similar, act9 și act10 corespund tranzițiilor între starea s1, sau respectiv s2 și starea nedorită CRASH. Acest lucru nu ar trebui să se întâmple în practică, deoarece o astfel de mișcare va rupe brațul.



FSM-ul de rotație (Adrian Turcanu & Talal Shaikh & Cristina Nicoleta Mazilu, 2020).

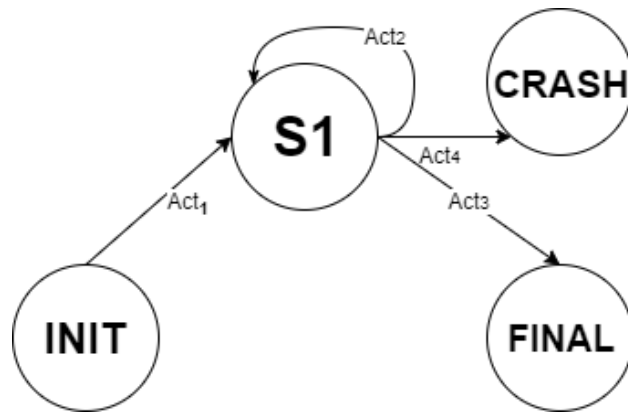
Componentele FSM-ului de translație sunt:

- alfabetul, $\Sigma = \{x\}$
- mulțimea de stări, $S = \{INIT, s1, FINAL\}$.
- starea inițială, $s0 = INIT$
- mulțimea de acțiuni, $ACT = \{act_i \mid 1 \leq i \leq 4\}$
- mulțimea de stări finale, $F = \{FINAL\}$.

Diagrama corespunzătoare este dată în figura următoare iar semnificațiile acțiunilor sunt după cum urmează:

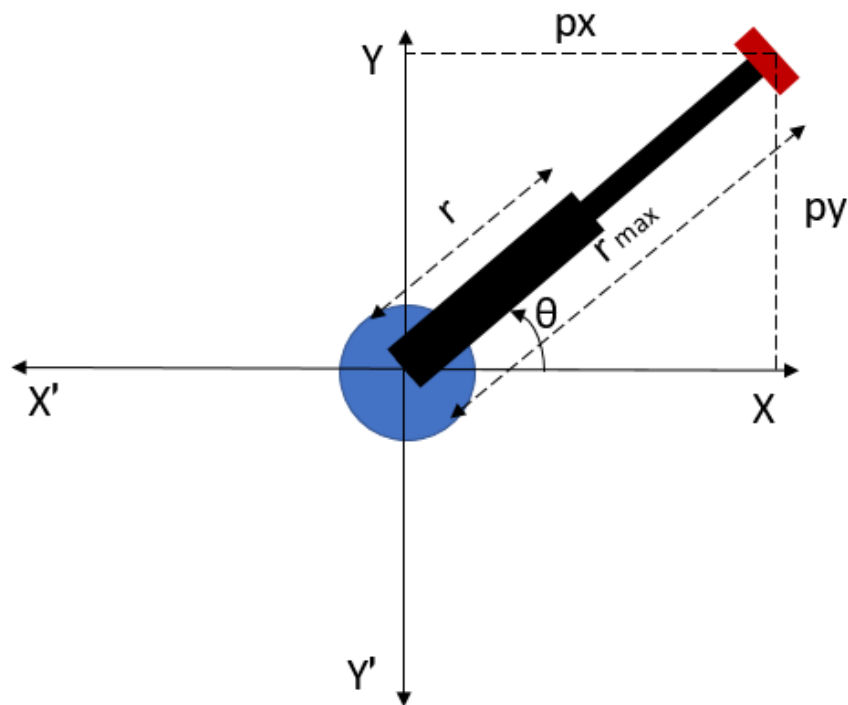
- act1 descrie mișcarea corespunzătoare extensiei brațului cu o valoare $t < MAX$. Prin urmare, starea se schimbă în s1.
- act2 definește mișcarea brațului până când nu a atins lungimea maximă. Prin urmare, starea este definită ca s1.

- act3 definește starea FINALĂ, ceea ce înseamnă că lungimea maximă a brațului este atinsă.
- act4 definește orice mișcare neobișnuită a brațului care depășește lungimea maximă permisă;
Prin urmare, starea CRASH este atinsă.



FSM-ului de translație (Adrian Turcanu & Talal Shaikh & Cristina Nicoleta Mazilu, 2020).

În figura de mai jos, se poate vedea o schiță a brațului robot, descriind mișcările de rotație și translație.



Vedere de sus a brațului (Adrian Turcanu & Talal Shaikh & Cristina Nicoleta Mazilu, 2020)

3.3 Un manipulator cu două articulații - studiu de caz

Această secțiune este dedicată studiului unui Model Event-B al cinematicii unui manipulator cu două articulații $r - \theta$. Pentru un astfel de mecanism, cinematica este simplă: o articulație se rotește în jurul trunchiului robotului cu un unghi θ și o articulație mișcă radial un braț de lungime r în interior și în exterior. Corectitudinea sintactică a Modelului este verificată utilizând unele dintre facilitățile Rodin, iar apoi unele proprietăți formulate utilizând LTL sunt verificate folosind verificadorul de Model ProB.

3.4 Modelul Event-B

Dezvoltarea Modelului Event-B se bazează pe metodologia descrisă în secțiunea 3.1. Prin urmare, Modelul se bazează pe un singur FSM de rotație și un singur FSM de translație. În consecință, Modelul Event-B conține un context și o mașină pentru fiecare dintre cele două FSM-uri. Contextul corespunzător rotației FSM, $C1$, conține setul de stări: $\{INIT, s1, s2, STOP, CRASH, interm\}$, unde starea *interm* este introdusă pentru simplitatea Modelării.

Modelul este descris în amănunt în teză.

3.5 Verificarea formală a Modelului

În această secțiune este demonstrată validarea Modelului prin utilizarea facilităților Rodin. În figura următoare se poate observa că toate constrângerile Modelului sunt generate și validate automat de sistem. Mai mult, toate stările au fost vizitate și nu există nicio încălcare a invariantilor atunci când se execută verificarea Modelului pe ambele mașini. Prin urmare, se face o validare primară a Modelului care poate fi îmbunătățită cu verificarea proprietăților LTL care descriu comportamentul manipulatorului. Unele dintre proprietățile Modelului sunt verificate și rezultatele sunt prezentate în teză.

- ▼ **M** RotationMachine
 - >  Variables
 - >  Invariants
 - >  Events
 - ▼  Proof Obligations
 -  INITIALISATION/inv2/INV
 -  INITIALISATION/inv3/INV
 -  INITIALISATION/inv4/INV
 -  INITIALISATION/inv5/INV
 -  INITIALISATION/inv6/INV
 -  INITIALISATION/inv7/INV
 -  INITIALISATION/act3/FIS
 -  move/inv3/INV
 -  move/inv4/INV
 -  move/inv6/INV
 -  move/inv7/INV
 -  move/act1/FIS
 -  move/act2/FIS
 -  changeStateToS1/inv2/INV
 -  changeStateToS1/inv5/INV
 -  reachStop/inv2/INV
 -  reachStop/inv5/INV
 -  changeStateToS2/inv2/INV
 -  changeStateToS2/inv5/INV
- ▼ **M** TranslationMachine
 - >  Variables
 - >  Invariants
 - >  Events
 - ▼  Proof Obligations
 -  INITIALISATION/inv2/INV
 -  INITIALISATION/inv3/INV
 -  INITIALISATION/inv4/INV
 -  INITIALISATION/act5/FIS
 -  translation/inv3/INV
 -  translation/inv4/INV
 -  translation/act1/FIS
 -  translation/act3/FIS
 -  changeStateToS1/inv2/INV

Validarea constrangerilor mașinilor de rotație și translație (Adrian Turcanu & Talal Shaikh & Cristina Nicoleta Mazilu, 2020)

Capitolul 4. Modele de cunoștințe pentru testarea software în robotică

Testarea software pe roboți reali poate fi complexă și costisitoare, prin urmare pot fi luate în considerare platformele de simulare, deoarece simulatoarele sunt mai rapide, mai ieftine și mai convenabile de utilizat. Dezvoltat de Cyberbotics Ltd, Webots este un software de simulare open source și multi-platformă (Webots, 2021), care este foarte popular în rândul cercetătorilor interesați de domeniul roboticii. Acest instrument a fost utilizat în această teză pentru generarea jurnalelor de evenimente corespunzătoare unui comportament simulat al unui robot de curățare. În plus, pentru implementarea Modelului de cunoștințe legat de comportamentul robotului, a fost utilizat limbajul standard de serializare a datelor YAML (YAML, 2021), deoarece acesta este integrat cu platforma Celonis.

Lucrarea constă în următoarele faze: (i) Modificarea codului robotului Webots adăugând detalii legate de acțiune și timp, pentru a obține jurnale de evenimente din comportamentul simulat (ii) Generarea Modelului de date pe baza acestor jurnale de evenimente folosind Celonis, (iii) Crearea Modelului de cunoștințe în Celonis EMS Studio, (iv) Crearea definițiilor relevante pentru KPI (Key Performance Indicator) în Modelul de cunoștințe v) Utilizarea KPI din Modelul de cunoștințe pentru testarea folosind mutații.

4.1 Metodologie

Metodologia începe cu asocierea stărilor fiecărui tip de mișcare a robotului (de exemplu, *înainte*, *înapoi*, *evitare obstacolul din stânga*, *evitare obstacolul din dreapta*). Fiecare caz este o colecție de activități și fiecare activitate este reprezentată de o mișcare. Fiecare caz trebuie să includă cel puțin următoarele informații: i) Identificatorul de caz, ii) Data și timpul, iii) Acțiunea iv) Numărul activității. Toate cazurile generate sunt stocate în jurnalele de evenimente, apoi acestea sunt conectate cu Celonis pentru obținerea Modelului de date. Mutații sunt apoi introduși în codul robotului și, în consecință, jurnalele de evenimente sunt regenerate folosind simulatorul. În cele din urmă, urmele originale și urmele mutantului sunt testate în raport cu Modelul de cunoștințe (KM) pentru a identifica apariția mutantului. Metodologia este demonstrată pe baza unui

Jurnalul de evenimente corespunzător activităților generate în Webots, după modificarea codului

C:\Users\adria\Documents\create\irobot test 2\worlds\create.wbt (irobot test 2) - Webots R2020b revision 1



Cele două KPI-uri din aceste Modele de cunoaștere sunt instrumente care utilizează limbajul de serializare YAML. Acestea corespund celor două stări descrise anterior și au fost obținute pe baza completării următoarelor caracteristici i) id, ii) descriere, iii) PQL, iv) format, v) direcție dorită.

Id-ul este unic și asigură consistența în întregul Model de cunoștințe, descrierea(*description*) conține detalii relevante cu privire la definiția KPI, *PQL* traduce definiția KPI-ului într-o interogare executabilă, *formatul* descrie tipul valorii numerice a indicatorului iar direcția (*desiredDirection*) indică comportamentul corect al KPI.

Utilizarea KPI-urilor definite pentru testarea folosind mutanți este detaliată în teză.

Capitolul 5. Minarea proceselor pe un mecanism robotizat

Acest capitol își propune să ofere o metodologie pentru verificarea conformității procesului unui mecanism robotizat. Metodologia este aplicată unui studiu de caz luând în considerare anumite urme operaționale sintetice care ar putea fi colectate prin tehnologiile Industrial Internet of Things (IIoT). Prin urmare, acest capitol sugerează o metodă practică de verificare a procesului real comparativ cu Modelul standard al procesului care ar putea fi furnizat de producătorul robotului. Studiul de caz al aplicării verificării conformității procesului este implementat folosind platforma de minare a proceselor - Celonis. În plus, vor fi utilizate, de asemenea, reprezentarea automatelor cu stări finite (FSM) și Modelarea proceselor (BPMN), deoarece se sugerează că aceste tehnici sunt adecvate pentru descrierea specificațiilor formale ale roboților.

5.1 Metodologie

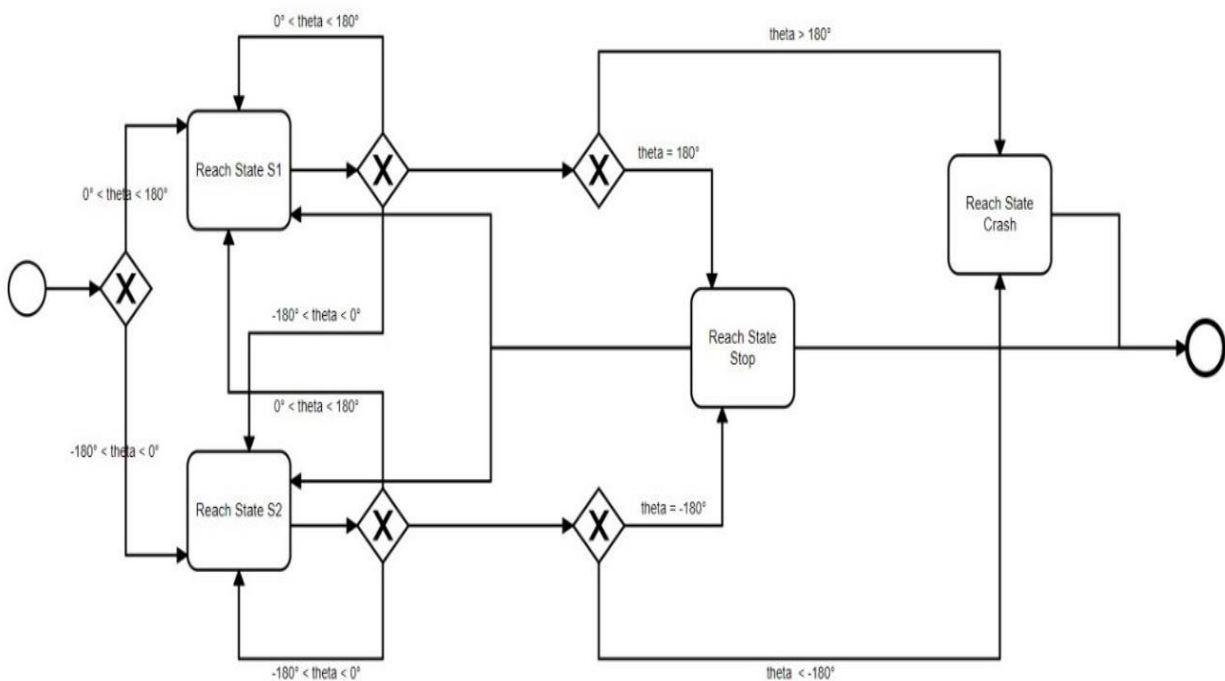
Descrise în cuvinte, pașii sunt după cum urmează: (i) un Model bazat pe FSM al unui mecanism robotizat este transformat într-un Model bazat pe BPMN 2.0, (ii) jurnalul de evenimente minat care conține procesul operațional al robotului este comparat cu procesul standard BPMN 2.0, (iii) Testarea conformității folosind Celonis trece automat prin fiecare caz în procesul operațional al jurnalului de evenimente și detectează toate încălcările procesului standard BPMN 2.0, (iv) în plus, Celonis permite opțiunea stabilirii regulilor pentru acțiuni inteligente determinate pentru încălcările detectate. Se sugerează că metodologia este utilă în diverse domenii industriale și poate oferi o imagine de ansamblu documentată a conformității procesului robotului și a modului în care informațiile despre proces pot fi transformate în acțiune. Metoda este demonstrată printr-un studiu de caz: un manipulator cu două articulații descris în Capitolul 3.

5.2 Experimentare

Punctul de plecare în acest caz este de a converti Modelul de rotație FSM al brațului robot din Capitolul 3 în modelul corespunzător BPMN 2.0. Următorul pas este analiza conformității procesului real creat utilizând jurnalul vs. modelul BPMN 2.0 creat anterior.

O abordare sistematică a fost utilizată pentru a obține modelul BPMN 2.0 din modelul corespunzător FSM. La convertirea FSM a mișcării de rotație în Modelul BPMN 2.0 (figura următoare), au fost utilizate unele porți exclusive atunci când un proces poate lua o singură cale din mai multe opțiuni, adică au fost utilizate pentru a stabili următoarele:

- Decizie cu privire la care dintre cele două stări S1 sau respectiv S2, fluxul de control va merge, în funcție de valoarea unghiului de rotație θ .
- Pentru a decide dacă rotația brațului cu unghiul θ trebuie să ajungă la starea STOP, în cazul în care valoarea $\theta = 180$ sau $\theta = -180$.



Modelul BPMN 2.0 corespunzător mișcării de rotație

Jurnalul de evenimente care conține urmele operaționale simulate ale mecanismului robotizat (în cazul mișcării de rotație) poate fi văzut în figura următoare. Acest jurnal de evenimente a fost

inserat în platforma dedicată minării proceselor – Celonis, pentru a efectua verificarea conformității procesului.

	A	B	C	D	E	F
1	Case ID	Activity	Timestamp	Angle theta current	Angle theta movement	
2	1000	Reach State S1	10/18/2020 9:00	0	179	
3	1000	Reach State S1	10/18/2020 9:01	179	-174	
4	1000	Reach State S2	10/18/2020 9:02	5	165	
5	1000	Reach State S2	10/18/2020 9:03	170	10	
6	1000	Reach State Stop	10/18/2020 9:04	180	-20	
7	1001	Reach State S1	10/19/2020 9:10	160	-60	
8	1001	Reach State S1	10/19/2020 9:11	100	20	
9	1001	Reach State S1	10/19/2020 9:12	120	50	
10	1001	Reach State S1	10/19/2020 9:13	170	10	
11	1001	Reach State Stop	10/19/2020 9:14	180	-122	
12	1002	Reach State S1	10/20/2020 9:20	58	-58	
13	1002	Reach State S1	10/20/2020 9:21	0	-5	
14	1002	Reach State S2	10/20/2020 9:22	-5	-75	
15	1002	Reach State S2	10/20/2020 9:23	-80	-90	
16	1002	Reach State S2	10/20/2020 9:24	-170	-10	
17	1002	Reach State Stop	10/20/2020 9:25	-180	150	
18	1003	Reach State S2	10/20/2020 9:26	-30	29	
19	1003	Reach State S2	10/21/2020 9:27	-1	-169	
20	1003	Reach State S2	10/21/2020 9:28	-170	110	
21	1003	Reach State S2	10/21/2020 9:29	-60	-121	
22	1003	Reach State Crash	10/21/2020 9:31	-181	31	
23	1004	Reach State S2	10/22/2020 9:16	-150	149	
24	1004	Reach State S2	10/22/2020 9:16	-1	51	
25	1004	Reach State S1	10/22/2020 9:20	50	-70	
26	1004	Reach State S2	10/22/2020 9:25	-20	20	
27	1004	Reach State S1	10/22/2020 9:30	0	181	
28	1004	Reach State Crash	10/22/2020 9:35	181	-211	
29	1005	Reach State S2	10/23/2020 9:26	-30	29	

Jurnalul evenimentelor de rotație

În cele ce urmează vor fi explicate detalii privind coloanele jurnalului de evenimente din figura de mai sus

Case ID - grupează activitățile conectate într-un caz.

Activity - stochează activitatea efectuată.

Timestamp - reprezentare data + ora pentru fiecare dintre activitățile efectuate.

Angle theta current (Unghiul curent theta) - este menit să indice poziția curentă a brațului.

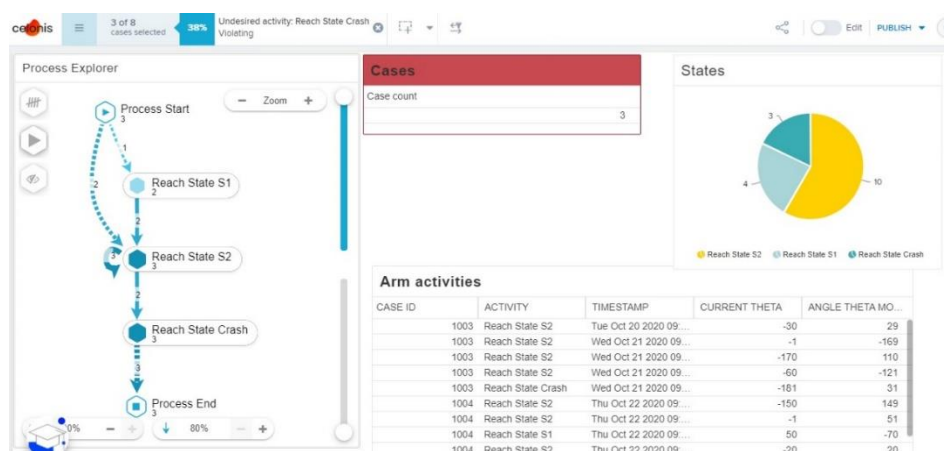
Angle theta movement (Mișcarea teta unghiulară) - este menită să indice unghiul de mișcare.

Din jurnalul de evenimente de mai jos se poate observa că în cazul procesului legat de ID-ul cazului 1003, la 20/10/2020 9:31 se ajunge la starea Crash din starea anterioară S2, când brațul trebuie să se deplaseze de la -60 grade cu încă -121 grade.

Similar sunt obținute jurnalele de evenimente și Modelul procesului BPMN 2.0 pentru mișcarea de translație (mai multe detalii se pot vedea în teză).

5.3 Verificarea conformității procesului de rotație

Una dintre cele mai importante caracteristici ale verificării conformității procesului folosind Celonis este identificarea încălcărilor procesului standard și diversele posibilități de analiză și acțiune asupra acestor cazuri. Figura următoare reprezintă o analiză personalizată (OLAP) care arată detalii din jurnalul de evenimente al mișcării de rotație care conține detalii ale cazurilor care se termină cu activitatea nedorită *Reach State Crash*, cum ar fi: numărul total de cazuri, vizualizarea procesului și a activităților, precum și o diagramă care prezintă detalii cu privire la aparițiile fiecărei stări.



Analize personalizate și tabele OLAP pentru rotație

Similar sunt obținute analize pentru mișcarea de translație (mai multe detalii se pot vedea în teză).

5.4 Transformarea Statisticilor procesului în acțiune folosind Celonis Action Engine

Action Engine este un instrument integrat în Celonis Intelligent Business Cloud de la sfârșitul anului 2018 (Badakhshan P. și colab., 2019).

Pornind de la analiza personalizată obținută anterior folosind tabelele OLAP, se poate crea un nou proiect Action Engine pentru o stare nedorită și pot fi definite unele semnale pentru a transmite incidente operatorilor. Fiecare semnal ar trebui să fie asociat cu unele reguli de rutare pentru a specifica mecanismele de redirectionare.

Action Engine funcționează ca un instrument RPA; prin urmare, monitorizează continuu datele procesului operațional în fundal pe baza setului de reguli definit și, atunci când este cazul, trimite alerte însoțite de acțiuni recomandate (Badakhshan P. și colab., 2019). Detalii se pot vedea în figurile următoare.

The screenshot shows the 'State Crash Alert' configuration page in the Celonis Action Engine. The page is titled 'State Crash Alert' and has a 'Save' button in the top right corner. The main content area is divided into two sections. The top section, 'How do you want to distribute Signals?', has three radio buttons: 'All Signals to same set of users', 'Distribute Signals based on a Data Model column' (which is selected), and 'Distribute Signals based on Signal attributes'. The bottom section contains three input fields: 'Rule Name' (State Crash Alert), 'Data Model' (Rotation (Rotation)), and 'Attribute Column' (Rotation_Event_Log.xlsx_Sheet1). Below these fields is a table with columns 'User', 'All', and 'Attributes'. The 'User' column shows 'You @ Cristina' and 'ct197@hw.ac.uk'. The 'All' column has a 'No' button. The 'Attributes' column has a series of buttons: '-181', '<-180', '<=-181', '=181', '>180', '>=181', and '181'. There is a 'Search users' input field and an 'Add User' button to the right of the table.

Reguli de rutare a motorului de acțiune

The screenshot shows the 'Alert State Crash' details page in the Celonis Action Engine. The page is titled 'Alert State Crash' and has a 'Status' dropdown menu. The main content area is divided into two sections. The top section, 'Please check this State Crash of 1007 that took place on Sun, 25 Oct 2020 09:35:00 and reached angle -181 and Stop the robotic mechanism', contains a table with columns 'CASE ID', 'ACTIVITY', 'TIMESTAMP', and 'CURRENT THETA'. The table has one row with values: '1007', 'Reach State Crash', 'Sun, 25 Oct 2020 09:35:00', and '-181'. Below the table is a section titled 'ANGLE THETA MOVEMENT' with the value '181'. The bottom section, 'Comments', has a text input field with the value 'Posting as Cristina' and a 'Send' button. To the right of the main content area is a sidebar with two sections: 'Actions' and 'Assignee'. The 'Actions' section has a button 'Please Stop The Robotic Arm Mechanism'. The 'Assignee' section shows 'You @ Cristina' and 'ct197@hw.ac.uk' with buttons 'Assign To Me' and 'Assign To User'.

Detalii despre incidentul unuia dintre State Crash identificat

Capitolul 6. Concluzii și activitatea viitoare

Teza sugerează și investighează câteva metodologii de testare și verificare a metodelor pentru sistemele robotizate. Se consideră fluxul de lucru de specificare formală și apoi verificare a proprietăților folosind Model Checking (Capitolul 3), se continuă cu specificarea și utilizarea unui sistem de gestionare a cunoștințelor (Knowledge Models) implementat pentru ca descrierea comportamentală a robotului să fie utilizată în faza de testare (a-priori la lansarea produsului) (Capitolul 4) și se termină cu utilizarea tehnicii de minare a procesului (Process Mining) în etapa de funcționare pentru detectarea în timp real a potențialelor neconformități. Pe langa minarea proceselor este investigată și o tehnică ce permite automatizarea acțiunilor de notificare pentru a rezolva problemele rapid și eficient datorită accesului la date (Capitolul 5).

Aceste metodologii ar trebui să permită inginerilor de sistem și software să efectueze verificarea Modelelor lor (Capitolul 3), să îmbunătățească eficiența testării (Capitolul 4), să îndeplinească cerințele de conformitate ale produselor în mod eficient și durabil (Capitolul 4).

Mai mult, prin utilizarea unui sistem de extragere a proceselor, companiile vor putea compara specificațiile produsului în raport cu performanțele operaționale în timp real ale robotului. Procedând astfel, s-ar putea asigura că mecanismul robotizat respectă specificațiile, prin înregistrarea detaliilor relevante din procesul operațional. În cazul în care există o abatere în procesul în timp real, sistemul de minare a procesului ar trebui să reacționeze la astfel de evenimente. Mai mult decât atât, acest sistem ar trebui să aplice analiza încălcării procesului, identificarea cauzei principale și acțiunile sugerate, pentru mai multă siguranță în toate operațiunile. Din momentul în care apare o abatere până la eliminare, rapoartele de neconformitate ar trebui să fie disponibile pentru a furniza cunoștințele valoroase și legăturile cu cauzele principale ale problemelor legate de sistemul robotizat. Metodologiile sugerate în această teză urmăresc să sugereze tehnici utile care pot fi utilizate pentru a preveni posibile probleme ale roboților dincolo de un pas desemnat.

6.1 Contribuții

Această teză a pornit de la investigarea cercetărilor existente privind verificarea Modelelor (Model Checking), testarea și validarea software folosind sisteme de gestionare a cunoștințelor

(Knowledge Management Systems) și verificarea conformității operaționale utilizând minarea proceselor (Process Mining) în domeniul roboticii. Au fost identificate lacune în literatura existentă constând în descrierea și aplicarea unor tehnici, metodologii și instrumente similare. Utilitatea metodologiilor prezentate de autoare pentru acest scop al tezei este demonstrată prin aplicarea lor pe studii de caz simple, dar relevante, care pot fi ușor extinse până la aplicabilitatea la industrial nivel.

6.2 Avantaje și limitări

Principalele avantaje ale utilizării Event-B și platformei Rodin pentru Model Checking, sunt simplitatea modelării, validarea automată a modelelor și facilitățile de verificare și animare a modelelor. Robustețea unui model Event-B este înrădăcinată în rigoarea limbajului matematic și în corectitudinea modelului care este asigurată în fiecare etapă a procesului de modelare prin diferite mecanisme automate. Mai mult, tehnica de rafinare permite utilizatorilor să dezvolte modele treptat și astfel să includă în modelul actual doar acele caracteristici esențiale în scopul etapei respective. Pe de altă parte, pot fi menționate ca limitare principală, tipurile de date numerice care sunt restricționate la subseturi de numere întregi.

Ca un avantaj al implementării sistemelor de gestionare a cunoștințelor (Knowledge Management) în domeniul testării software, se poate menționa faptul că inginerii și testerii software sunt obișnuiți cu diverse instrumente și sisteme IT. Ca o provocare, se poate menționa necesitatea unei forțe de muncă dedicate, capabilă să utilizeze eficient un asemenea sistem.

În contextul Industriei 4.0, folosind sisteme Cloud pentru a permite procesarea centralizată a datelor, deoarece minarea proceselor (Process Mining) este o soluție bazată pe Cloud, această tehnică ar putea fi adoptată pentru prevenirea posibilelor incidente în timpul operațional al mecanismelor robotizate.

Cu toate acestea, deși Industria 4.0 împuternicește sistemele de fabricație cu roboții industriali, configurarea, implementarea, monitorizarea și întreținerea dispozitivelor necesare și a datelor mari rezultate generează unele sarcini dificile, în special datorită naturii complexe a cantității massive de date colectate de la senzori.

O mare provocare în adoptarea verificării conformității proceselor în timp real pentru roboții industriali este capacitatea de a selecta eficient din cantitatea uriașă de date brute generate, cea care este cu adevărat relevantă din perspectiva procesului de extragere a datelor și de a o alinia la cerințele unui jurnal de evenimente. În plus, conexiunea și integrarea datelor sunt o altă provocare majoră în implementarea extragerii informațiilor pentru obținerea vizualizării în timp real a fluxului procesului și pentru detectarea anomaliilor. Compania care dorește să implementeze o asemenea metodologie, trebuie să poată conecta un număr mare de dispozitive diferite și să susțină transferul de date cu o latență scăzută. Un alt aspect care trebuie luat în considerare este costul ridicat al integrării tuturor acestor sisteme, care necesită software, hardware, echipamente noi și forță de muncă instruită.

Deși există unele provocări cu adoptarea minării proceselor, multe companii din diferite industrii adoptă astfel de soluții, permițând colectarea și analiza unei cantități mari de date pentru a deschide calea către o performanță mai bună, o simplificare a operațiunilor și reducerea costurilor generale.

6.3 Activitatea viitoare

Viitoarele lucrări se vor concentra pe aplicarea unei metodologii similare de verificare a modelelor pentru alte tipuri de mecanisme robotizate, cu mai multă complexitate și mai multe capacități. Mai mult, planul este de a investiga și alte limbaje și platforme de modelare care ar putea ajuta la depășirea unora dintre limitările actuale ale Event-B și Rodin.

Din direcția de cercetare în domeniul gestionării cunoștințelor, viitoarele lucrări se vor concentra pe dezvoltarea unui Model de cunoștințe complet pentru testarea software a roboților folosind diferite instrumente și tehnologii, cum ar fi cele care permit vizualizări interactive. Pe de altă parte, explorarea capacităților și tehnicilor de gestionare a cunoștințelor aplicate diferitelor tipuri de roboți și mecanisme robotizate este unul dintre principalele domenii de interes legate de acest domeniu.

Perspectiva în direcția cercetării legată de minarea proceselor, este de a efectua în continuare studii care ar putea aduce o anumită valoare adăugată industriei roboților și a mecanismelor robotizate. Planul este continuarea utilizării minării proceselor roboților combinată cu alte tehnologii de

ultimă generație. Principalul scop viitor legat de această metodologie este acela de a încerca aplicarea acestor metodologii pe diferite tipuri de roboți industriali și de a obține colaborări în cadrul industriilor și organizațiilor conexe.

Bibliografie

- A Mammar, M Frappier (2020) Modeling of a speed control system using Event-B. International Conference on Rigorous State-Based Methods 12071: 367-381.
- A Mammar, R Laleau (2015) Modeling a landing gear system in Event-B. International Journal on Software Tools for Technology Transfer 19: 167-186.
- A Turcanu, L Mierla, F Ipaté, et al. (2014) Modelling and analysis of *coli* respiratory chain. Applications of Membrane Computing in Systems and Synthetic Biology 7: 247-266.
- A. Engels, L. Feijs, and S. Mauw (1997). Test generation for intelligent networks using Model checking. In Proceedings of TACAS97, volume 1217 of LNCS, pages 384–398. Springer.
- A. Turcanu and F. Ipaté (2011). Modelling, testing and verification of P systems with active membranes using Rodin and ProB. In Proc. CMC12, pages 459–468. Paris-Est University Press.
- A. Turcanu and F. Ipaté (2012). Computational properties of two P systems solving the 3-colouring problem. In Proc. SYNASC12, pages 62–70. IEEE Computer Society.
- A. Turcanu and F. Ipaté (2012). Simplifying Event-B Models of P systems using functions. In Proc. CMC13, pages 455–458. Computer and Automation Research Institute of the Hungarian Academy of Sciences.
- Aagesen, Gustav & Krogstie, John. (2015) BPMN 2.0 for Modeling Business Processes. Handbook on Business Process Management 1: Introduction, Methods, and Information Systems. 219-250. 10.1007/978-3-642-45100-3_10.
- Aalst, Wil & Reijers, Hajo & Weijters, A. & Dongen, Boudewijn & Medeiros, Ana & Song, Minseok & Verbeek, H. (2007). Business process mining: an industrial application. J Inf Syst. Inf. Syst. 32. 713- 732.
- Abrial JR (2007) A system development process with Event-B and the rodin platform. Butler M, Hinchey MG, Larrondo-Petrie MM, Formal Methods and Software Engineering. (edn), ICFEM 2007, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg 4789.

- Adrian Turcanu & Talal Shaikh & Cristina Nicoleta Mazilu (Turcanu) (2020). On Model Checking of a Robotic Mechanism. *Journal of Robotics and Automation*. Volume 4 | Issue 1 | DOI: 10.36959/673/366
- Aichernig B.K., Mostowski W., Mousavi M.R., Tappler M., Taromirad M. (2018). Model Learning and Model-Based Testing. In: Bennaceur A., Hähnle R., Meinke K. (eds) *Machine Learning for Dynamic Software Analysis: Potentials and Limits*. Lecture Notes in Computer Science, vol 11026. Springer, Cham. https://doi.org/10.1007/978-3-319-96562-8_3
- Anca Ioana ANDREESCU & Iulian INTORSUREANU & Rodica MIHALCA & Adina UTA, (2008). Knowledge Management in E-Learning Systems, *Informatica Economica*, Academy of Economic Studies - Bucharest, Romania, vol. 0(2), pages 60-65.
- B. Falah, S. Bouriat, O. Achahbar (2015). Reducing Mutation Testing Cost Using Random Selective Mutation Technique, *Malaysian Journal of CS*. Vol. 28(4)
- Badakhshan, P. et al. (2019) The Action Engine – Turning Process Insights into Action. In: 1st International Conference on Process Mining, Aachen, Germany. <http://ceur-ws.org/Vol-2374/paper8.pdf>.
- Balogh Richard, David Obdrzalek (2018). Using finite state machines in introductory robotics. *International Conference on Robotics and Education RiE 2017* 829: 85-91.
- Brugali D, Cisternino G, Colombo D, et al. (2007). Trends in robotic software frameworks. *Software Engineering for Experimental Robotics*, 259-266.
- Business Process Model and Notation (BPMN) (2020), Version 2.0, <https://www.omg.org/spec/BPMN/2.0/>, Accessed 11/2020.
- C Zielinski, M Figat, R Hexel (2018). Robotic systems implementation based on FSMs, *Conference on Automation* 742: 441-452.
- Celik, Ufuk & Akçetin, Eyüp (2018). Process Mining Tools Comparison. 9. 97-104. 10.5824/1309-1581.2018.4.007.x.
- Celonis (2021), Celonis, <https://www.celonis.com/>, Accessed 05/2021

- Cristina Nicoleta Turcanu (2021). Process Mining on a Robotic Mechanism. IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2021, pp. 205-212, doi: 10.1109/ICSTW52544.2021.00043.
- Cuena, J., & Molina, M. (2000) The role of knowledge Modelling techniques in software development: a general approach based on a knowledge management tool, Int. J. Hum. Comput. Stud., 52, 385-421.
- Dayan, Rony & Evans, Stephen. (2006). KM your way to CMMI. J. Knowledge Management. 10. 69-80. 10.1108/13673270610650111.
- DEPLOY website (2020) Industrial deployment of system engineering methods, <http://deploy-eprints.ecs.soton.ac.uk/view/type>, Accessed 02/2020.
- Douglass Bruce Powel (1999). Doing hard time: Developing real-time systems with UML, objects, frameworks, and patterns, Addison Wesley, 749.
- Duffy, J. (2000). Knowledge management: to be or not to be ?, Information Management Journal, Vol. 34 No. 1, pp. 64-67.
- Edmund M Clarke, Orna Grumberg, Doron A Peled (1999) Model checking. The MIT Press.
- Edmund M Clarke, William Klieber, Milos N, et al. (2011) Model checking and the state explosion problem. LNCS series 7682.
- Érica Ferreira de Souza, Ricardo de Almeida Falbo, Nandamudi L. Vijaykumar (2015) Knowledge management initiatives in software testing: A mapping study, Information and Software Technology, Volume 57, Pages 378-39.
- Event-B website (2020), <http://www.event-b.org/>, Accessed 03/2020
- F Ipaté, A Turcanu (2012). Modeling, verification and testing of P Systems using rodin and ProB. Proceedings of CMC12: 12 th International Conference of Membrane Computing.
- Figat Maksym, Cezary Zielinski, Rene Hexel (2017). FSM based specification of robot control system activities. 2017 11th International Workshop on Robot Motion and Control (RoMoCo)

- HeeSun Choi, Cindy Crump, Christian Duriez, Asher Elmquist, Gregory Hager, David Han, Frank Hearl, Jessica Hodgins, Abhinandan Jain, Frederick Leve, Chen Li, Franziska Meier, Dan Negrut, Ludovic Righetti, Alberto Rodriguez, Jie Tan, Jeff Trinkle (2021). On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences* Jan, 118 (1) e1907856118; DOI: 10.1073/pnas.1907856118
- Hoang Thai Son, Snook Colin, Asieh Salehi, et al. (2018). Validating and verifying the requirements and design of a haemodialysis machine using the rodin toolset. *Science of Computer Programming* 158: 122-147.
- Hoang, Thai Son. (2013). An Introduction to the Event-B Modelling Method, In book: *Industrial Deployment of System Engineering Methods* (pp.211-236) Publisher: Springer-Verlag
- Hong Seong and Jeong Seok (2012). SITAF: simulation-based interface testing automation framework for robot software component, , *Automation InTech*
- IDEF-3x (1993). Process Modeling language specification, standard NA-94-1422B, Rockwell International
- J. Callahan, F. Schneider, and S. Easterbrook (1996). Automated Software Testing Using Model-Checking. In *Proceedings of SPIN Workshop*.
- Jarrar Abdessamad, Balouki Youssef (2018). Towards sophisticated air traffic control system using formal methods, *Modelling and Simulation in Engineering*.
- K. Kumar and P. S. Reel (2011). Analysis of contemporary robotics simulators, 2011 International Conference on Emerging Trends in Electrical and Computer Technology, pp. 661-665.
- K. Wnuk and T. Garrepalli (2018). Knowledge management in software testing: A systematic snowball literature review, *e-Informatica Software Engineering Journal*, vol. 12, no. 1. Politechnika Wroclawska, pp. 51–78.
- Kocbek Bule, Mateja & Jost, Gregor & Hericko, Marjan & Polančič, Gregor (2015). Business Process Model and Notation: The Current State of Affairs. *Computer Science and Information Systems*. 12. 509- 539. 10.2298/CSIS140610006K.

- Koenig, M.E.D. (2012). What is KM? Knowledge management explained, KM World, May, 2012, available at: www.kmworld.com/Articles/Editorial/What-Is-.../What-is-KM-Knowledge-Management-Explained-82405.aspx (Accessed 15 August 2021).
- M. Alavi and D.E. Leidner (2001), MIS Quarterly, 25 (1), pp. 107-136
- M. González-Loureiro, M. Vila and G. Schiuma (2015). Investigaciones Europeas de Dirección y Economía de la Empresa, 21 (2), pp. 55-57
- Markus Weißmann, Stefan Bedenk, C. Buckl, A. Knoll less (2011). Model Checking Industrial Robot Systems. Published in SPIN, Computer Science
- McIntyre, SG, Gauvin, M & Waruszynski, B (2003). Knowledge management in the military context, Canadian Military Journal, Spring, 35-40.
- Mealy George H (1955). A method for synthesizing sequential circuits. The Bell System Technical Journal 34: 1045-1079.
- Mery Dominique, Singh Neeraj (2011) Automatic code generation from event-B Models, ACM International Conference Proceeding Series.
- Michael Brady (Editor), L.A. Gerhardt (Editor), H.F. Davidson (Editor) (1984). Robotics and Artificial Intelligence (Nato ASI Subseries F: (11)) Softcover reprint of the original 1st ed.
- Michel O. (2004). Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation, International Journal of Advanced Robotic Systems.
- Moore Edward F (1956). Gedanken-experiments on sequential machines. In: CE Shannon, J McCarthy, Automata Studies. (edn), Annals of Mathematical Studies, Princeton University Press.
- Mora, Antonio & Merelo Guervós, Juan & García-Sánchez, Pablo & Castillo, Pedro & Rodríguez-Domingo, Maria & Hidalgo-Bermúdez, Rosa. (2014) Creating autonomous agents for playing Super Mario Bros game by means of evolutionary finite state machines, Evolutionary Intelligence. 6. 1-14. 10.1007/s12065-014-0105-7.
- Muşat F., Mişu F. (2018) Collaborative Robots and Knowledge Management - A Short Review. Acta Universitatis Cibiniensis. Technical Series.

- Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organ. Sci.* 5, 14–37.
doi: 10.1093/fampra/cm098
- Nonaka, I., and Takeuchi, K. (1995). *The Knowledge Creating Company*. New York, NY: Oxford University Press.
- Noy, N. & McGuinness, Deborah. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Knowledge Systems Laboratory. 32.
- P. Drucker (1985) *Innovation and entrepreneurship: practice and principles*, Harper & Row, New York.
- Paul E Black (2016) *Dictionary of algorithms and data structures*. NIST.
- ProB website (2020) What is ProB, <https://prob.hhu.de/>, Accessed 03/2020.
- Rahul Karmakar, Bidyut Biman Sarkar, Nabendu Chaki (2020). Event-B based formal Modeling of a controller: A case study. *EasyChair*.
- Rocha, Álvaro; Correia, Ana Maria; Adeli, Hojjat; Reis, Luís Paulo; Teixeira, Marcelo Mendonça (2016) New advances in information systems and technologies, *Advances in Intelligent Systems and Computing*, Vol. 445.
- Rojas, Eric & Munoz-Gama, Jorge & Sepulveda, Marcos & Capurro, Daniel (2016) Process Mining in Healthcare: A literature review, *Journal of biomedical informatics*. 61. 10.1016/j.jbi.2016.04.007.
- Roldán, J.J., Olivares-Méndez, M.A., del Cerro, J. et al (2018), Analyzing and improving multi-robot missions by using process mining. *Auton Robot* 42, 1187–1205
<https://doi.org/10.1007/s10514-017-9686-1>
- Roman A., Mnich M. (2021) Test-driven development with mutation testing -an experimental study. *Software Qual J* 29, 1–38.
- Rozinat A., van der Aalst W.M.P. (2006) Decision Mining in ProM. In: Dustdar S., Fiadeiro J.L., Sheth A.P. (eds) *Business Process Management. BPM 2006. Lecture Notes in Computer Science*, vol 4102. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11841760_33

- Rus, Ioana. (2002). Guest Editors' Introduction: Knowledge Management in Software Engineering. *IEEE Software*. 19. 26-38. 10.1109/MS.2002.1003450.
- S. Hamimoune, B. Falah (2016). Mutation testing techniques: A comparative study, 2016 International Conference on Engineering & MIS (ICEMIS).
- Saint-Onge, Hubert (1996). Tacit knowledge: the key to the strategic alignment of intellectual capital, *Strategy & Leadership*, vol. 24, no. 2, Mar.-Apr. 1996, p. 10-16.
- Shriyam, Shaurya & Gupta, Satyandra. (2019). Modeling and verification of contingency resolution strategies for multi-robot missions using temporal logic. *International Journal of Advanced Robotic Systems*. 16. 172988141988569. 10.1177/1729881419885697.
- SN Predut, F Ipat, M Gheorghe, et al. (2018) Formal Modelling of cruise control system using Event-B and rodin platform. *IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems(HPCC/SmartCity/DSS)*, 1541-1546.
- T. Tammet, E. Reilent, M. Puju, A. Puusepp, A. Kuusik (2010) Knowledge Centric Architecture for a Robot Swarm, *IFAC Proceedings Volumes*
- Tseng, S.M. (2008) Knowledge management system performance measure index, *Expert Systems with Applications*, Vol. 34 No. 1, pp. 734-745.
- Turcanu A, Shaikh T, Mazilu CN (2020) On Model Checking of a Robotic Mechanism. *J Robotics Autom* 4(1):158-165.
- U Tikhonova, M Manders, M Brand, et al. (2013) Applying Model transformation and Event-B for specifying an industrial DSL. *CEUR Workshop Proceedings*, 41-50.
- Van der Aalst WMP, Desel J, Oberweis A (2000) Business process management. *Lecture notes in computer science*, vol 1806. Springer, Berlin/Heidelberg/New York
- Vanderson HF, Adenilso S, Reza MM (2019) Hierarchical featured state machines. *Science of Computer Programming* 171: 67-88.

- Víctor Rivera, Néstor Cataño, Tim Wahls, et al. (2015) Code generation for Event-B. *International Journal on Software Tools for Technology Transfer* 19: 31-52.
- W.Z. Khan, M.H. Rehman, H.M. Zangoti, M.K. Afzal, N. Armi, K. Salah (2020) Industrial internet of things: Recent advances, enabling technologies and open challenges. *Computers & Electrical Engineering*, Volume 81, ISSN 0045-7906
- Waldron K, Schmiedeler J (2008) *Kinematics*. Springer Handbook of Robotics. Springer, Berlin, Heidelberg.
- Wang, J., Ding, D., Liu, O. and Li, M. (2016) A synthetic method for knowledge management performance evaluation based on triangular fuzzy number and group support systems, *Applied Soft Computing*, Vol. 39, pp. 11-20.
- Wang, K.Y., Tan, L.P., Cheng, S.L. and Wong, W.P. (2015) Knowledge management performance measurement: measures, approaches, trends and future directions, *Information Development*, Vol. 31 No. 3.
- Wang, Y. and Zheng, J. (2010) Knowledge management performance evaluation based on triangular fuzzy number, *Procedia Engineering*, Vol. 7 No. 8, pp. 38-45.
- Webots (2021) Cyberbotics, <http://www.cyberbotics.com>, Accessed 05/2021
- Wiig, K. (1999) *Comprehensive Knowledge Management - Working Paper*, Knowledge Research Institute Inc., http://www.knowledgeresearch.com/downloads/compreh_km.pdf, Accessed August 2021
- Wil M.P. van der Aalst (2011) *Process Mining Discovery, Conformance and Enhancement of Business Processes*, Springer-Verlaminig
- Wil Van der Aalst W. et al. (2012) *Process Mining Manifesto*. In: Daniel F., Barkaoui K., Dustdar S. (eds) *Business Process Management Workshops. BPM 2011. Lecture Notes in Business Information Processing*, vol 99. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-28108-2_19

- Wu, Y.L., Wang, X. and Wu, H.S. (2009) Research on the performance measurement of knowledge management based on principal component analysis, *International Workshop on Intelligent Systems and Applications*, IEEE, pp. 1-4.
- Yamine Ait Ameer, Idir Ait Sadoune, Kahina Hacid, et al. (2018) Formal Modelling of ontologies: An Event-B based approach using the rodin platform. *Electronic Proceedings in Theoretical Computer Science* 271: 24-33.
- L. V. Gómez, J. Miura (2021) Ontology-based knowledge management with verbal interaction for command interpretation and execution by home service robots, *Robotics and Autonomous Systems*, Volume 140, ISSN 0921-8890