Raport ştiinţific și tehnic

privind implementarea proiectului **MACPS** în perioada iulie 2017 –decembrie 2019

Scopul proiectului este realizarea unei abordări integrative pentru specificarea, analiza și simularea Sistemelor Cyber-Physical. Aceasta va consta dintr-o metodă pe trei nivele: (a) o bază teoretică nouă pentru modelarea și verificarea Sistemelor Cyber-Physical care acționează și interacționează cu scenarii de mediu cu incertitudine și chiar necunoscute; (b) o abordare integrată de verificare și testare formală pentru aceste sisteme; (c) un ansamblu de utilitare care să permită specificarea, analiza și simularea.

În acest scop sunt investigate două paradigme de modelare care vor fi folosite în contextul Sistemelor Cyber-Physical: (1) X-maşinile / mașinile cu stări finite extinse și (2) sistemele de membrane (sistemele P). Aceste două formalisme de modelare au capabilități de modelare complementare și și-au dovedit adecvarea pentru specificarea și analiza unor clase de aplicații complexe.

1. Principalele rezultate obţinute

Pentru atingerea obiectivelor mai sus menţionate, în perioada iulie 2017 – decembrie 2019, se remarcă realizări importante în următoarele direcţii principale:

- Dezvoltarea de metode de învățare din întrebări și contraexemple pentru X-mașini, ca o generalizare a metodelor existente pentru automate finite (algoritmi de tip Angluin).
- Investigații privind folosirea sistemelor kP pentru modelare și validare (în aplicații inginerești, în speță sistemul de cruise control al unei biciclete electrice, algoritmi de sortare și probleme de broadcasting, precum și pentru modelarea altor clase de sisteme P).
- Dezvoltarea unei abordări de generare de seturi de test bazate pe mașini cu stări finite extinse și X-mașini folosind algoritmi genetici multi-obiectiv.
- Studiul proprietății de accesibilitate (reachability) a unei clase de sisteme parametrizate folosind o metodă numită “regular model checking”.
- Testare Model-in-the-Loop pentru Sisteme Cyber Physical
- Modelare și verificare formală folsind Event-B și platforma Rodin

Fiecare dintre direcţiile mai sus menţionate vor fi trecute în revistă în ceea ce urmează, evidenţiindu-se realizările principale.

1.1. Dezvoltarea de metode de învățare din întrebări și contraexemple pentru X-mașini

Noțiunea de stare este centrală pentru analiza și verificarea formală și pentru testarea bazată pe model. Majoritatea modelelor construite pentru acest scop sunt bazate pe mașini cu stări finite. Pe de altă parte, în practică un model formal al sistemului poate să nu existe, și chiar când acesta există poate să nu mai fie actual datorită schimbărilor frecvente în specificație sau a comportamentului necunoscut/incert care poate apărea. Un mod de a face reverse engineering pentru un model bazat pe stări este prin învățare din întrebări. Învățarea mașinilor cu stări finite din întrebări a fost introdusă de Angluin, care a propus și un algoritm, numit L*. În acest scenariu, un learner pune întrebări și un teacher furnizează răspunsuri. Algoritmul L* inferează un limbaj regulat, în forma unui automat determinist, din răspunsurile la o mulțime finită de întrebări de apartenență și întrebări de echivalență. O întrebare de echivalență întreabă dacă o anumită secvență de intrare este acceptată de către sistem. Alături de întrebări de apartenență, L* folosește întrebări de echivalență pentru a verifica dacă algoritmul de învățare s-a terminat. Oracolul de echivalență funizează contraexemple dacă automatul construit din informația disponibilă pană acum nu corespunde limbajului dat. Gold a arătat că gasirea unui automat determinist minimal consistent cu o mulțime arbitrară de exemple pozitive și negative este NP-hard. Algoritmul de învățare are avantajul de a putea să selecteze exemple pentru întrebările de apartenență, deci mulțimea de exemple pentru întrebările de apartenență nu este arbitrară. Folosind în plus și întrebări de echivalență, algoritmul L* poate învăța automate finite în timp polynomial în numărul lor de stări.

Totuși, construirea unui model bazat pe stări a sistemului, și deci folosirea metodelor de verificare formală și testare bazată pe un astfel de model, suferă de bine-cunoscuta problemă a exploziei stărilor. Această problema este foarte prezentă când avem de-a face cu Sisteme Cyber-Physical, când un număr mare de echipamente fizice și algoritmi de calculator pot produce un număr nefezabil de mare de stări ale modelului care este învățat. O soluție pentru această problemă este folosirea unor mașini cu stări finite extinse, care folsesc, alături de structura de stări, care modelează controlul sistemului, o structură de memorie, folosită pentru modelarea datelor. O astfel de mașină cu stări finite extinsă este X-mașina (comunicantă), care furnizează o definiție precisă atât a comportamentului individual cât și a celui colectiv specificând stările, memoria internă, funcțiile (care reprezintă tranzițiile dintre stări) și protocoalele de comunicație. O combinație de diagrame de stare și fluxuri de date, foarte asemanătoare cu X-mașinile, a fost folosită cu mult succes pentru specificarea comportamentului unor variate sisteme

auto. Un alt avantaj important al X-mașinilor este existenta unor metode de testare asociate: acestea garantează corectitudinea implementării și definește cerințe constructive, numite conditii de “design pentru test”, pe care un sistem trebuie să le satisfacă pentru a fi testabil. Aplicând aceste metode, un sistem software complex este descompus într-o ierarhie de X-mașini, proiectat într-un mod top-down și testat într-un mod bottom-up.

Extinderea algoritmilor existenți pentru automate finite la X-mașini nu este imediată, datorită caracteristicilor suplimentare (structura de memorie și funcțiile de procesare). În articolul [2] este investigată problema învățării X-mașinilor pentru cele doua situații principale ale condițiilor de „design pentru test”. Este arătat că, pentru condiții de „design pentru test” mai restrictive, învățarea X-mașinilor se poate reduce la algoritmul L*, folosindu-se în plus un constraint solver pentru determinarea fezabilității sevențelor de funcții de procesare. În cazul mai general al condițiilor de „design pentru test” mai puțin restrictive, este necesar un algoritm mai complex, de separare a două limbaje regulate. În acest algoritm este învățat un automat finit care acceptă toate secvențele dintr-un prim limbaj U1 și rejectează secvențele dintr-un al doilea limbaj U2, putând avea orice comportament pentru restul secvențelor. În acest al doilea caz al condițiilor de „design pentru test” este folosită o X-mașină cu trei tipuri de stări (accept, reject și don’t care). Pentru realizarea algoritmului de învățare, este dezvoltată baza teoretică pentru determinarea unei X-mașini minimale consistente cu o X-mașină cu trei tipuri de stări și este propus un algoritm pentru aceasta.

1.2. Investigații privind folosirea sistemelor kP pentru modelare și validare

Unul dintre cele mai naturale moduri de a descrie interacțiunea component-component (inspirată din biologie, dar cu largi aplicații) este furnizată de calculul cu membrane. Inspirate din reacțiile biochimice din celule, sistemele de membrane (numite și sisteme P) sunt una dintre puținele abordări existente care combină aspecte cantitative, calitative și topologice cu un mecanism flexibil pentru a capta paralelismul inerent al rectiilor biochmice. Sistemele de membrane au fost extinse, de exemplu la sisteme țesut și sisteme de populații pentru a capta trăsăturile organismelor celulare și ale populațiilor de celule.

În ultimii ani, s-au făcut progrese semnificative în modelarea și simularea sistemelor din domenii variate. Deși acest model este inspirat din biologie, mulțimea de aplicații acestui paradigm computațional merge mult dincolo de această clasă de sisteme, arătându-și potențialul pentru specificarea de sisteme cu un grad mare de complexitate. Pentru a facilita modelarea, în multe cazuri trăsături variate au fost adăugate într-o manieră ad-hoc acestor clase de sisteme P. Aceasta a dus la o multitudine de variante de

sisteme P, fără un mod coerent de integrare. Conceptul de sistem P nucleu (kernel P system, sistem kP) furnizează un răspuns la această problemă.

Un sistem kP integrează într-o manieră coerentă și elegantă multe dintre trăsăturile sistemelor P folosite cu mare succes în modelare și, în consecință, furnizează un cadru pentru analiza acestor modele. Expresivitatea acestor sisteme a fost ilustrată de un număr de studii de caz reprezentative. Modelul Sistemelor kP este susținut de un limbaj de modelare, numit kP-Lingua, capabil de a transforma specificații sub formă de sisteme kP într-o reprezentare ce poate fi citită automat de un program. Mai mult, există un software framework, kPWorkbench, care integrează o mulțime de tehnici și utilitare de simulare și verificare.

Articolul [3] ilustrează folosirea sistemelor kP pentru modelarea și validarea unor aplicații inginerești, în speță sistemul de cruise control al unei biciclete electrice. Validarea sistemului este demonstrată prin verificare formală efectuată folosind utilitarul kPWorkbench. În plus, se arată că modelul de forma unui sistem kP poate fi testat folosind metode bazate pe X-mașini și automate finite.

Articolul [1] ilustrează capabilitățile de modelare ale sistemelor kP, arătând cum pot fi reprezentate prin acest formalism alte clase de sisteme kP și furnizând un număr de modele de forma unor sisteme kP pentru un algoritm de sortare și o problemă de broadcasting. De asemenea se arată cum poate fi folosită verificarea formală pentru a arăta că modelele funcționeaza conform scopului. În final, o metodă de generare de teste bazată pe automate finite este extinsă pentru sisteme kP nedeterministe.

Articolele [16, 17, 18] prezintă o metodă de generare de date de test ce reprezintă valori de intrare pentru modelele unor tipuri speciale de sisteme kP. Lucrarea [16] prezintă ideile și experimentele preliminare care au stat la baza implementării algoritmului de testare. Lucrarea [17] conține prezentarea detaliată a algoritmului de generare de teste. De asemenea, este introdus modelul de sisteme kP, care poate fi asemănat cu o mașină cu stări finite. Noul model de sistem kP, conține un compartiment care simulează transmiterea de intrări sistemului. Având la intrare un sistem kP și un set de pași de execuție al sistemului, această metodă generează, folosind algoritmi genetici, un set de valori care pot fi transmise compartimentului principal de către compartimentul de intrări, astfel încât rularea sistemului să determine exact pașii de execuție primiți ca intrare. Lucrarea [18] extinde articolul [17].

1.3. Generare de seturi de test bazate pe mașini cu stări finite extinse folosind algoritmi genetici multi-obiectiv

Folosirea mașinilor cu stări finite extinse / X-mașinilor pentru generare de date de test poate fi un proces dificil pentru că trebuie să generăm căi fezabile prin model și, în plus, trebuie să găsim date de

intrare care traversează aceste căi. Articolul [5] prezintă un algoritm de generare de seturi de date de test pentru mașini cu stări finite extinse. Algoritmul produce un set de căi fezabile care acoperă toate tranzițiile modelului folosind un algoritm genetic cu funcție multi-obiectiv (modificat prin ștergerea căilor redundante și reducerea lungimii căilor). Funcția multi-obiectiv are ca scop optimizarea acoperirii tranzițiilor și fezabilității căilor, pe baza dependențelor fluxului de date. Având un set de căi rezultate în urma aplicării algoritmului, problema găsirii parametrilor de intrare pentru fiecare cale este relativ facilă. Metoda prezentată poate fi de asemenea aplicată în cadrul similar al X-mașinilor.

Articolul [19] prezintă un algoritm de generare de seturi de date de test pentru mașini cu stări finite extinse. Algoritmul produce un set de căi fezabile care acoperă toate tranzițiile modelului folosind un algoritm genetic cu funcție multi-obiectiv. Căile generate cu această metodă s-au dovedit a fi mai complexe decât cele generate cu metoda prezentată în lucrarea [5], combinând metoda de selectare a căilor cu cea prezentată în lucrarea [8]. În plus, noua abordare folosește algoritmul NSGA-III, obținând rezultate mai bune decât alți algoritmi genetici pentru această problemă. Funcția multi-obiectiv folosită are ca scop optimizarea acoperirii tranzitiilor si a fezabilitatii căilor, urmărind și maximizarea diferențelor dintre tranzițiile căilor.

Articolul [20] prezintă o metodă ce combină algoritmii prezentați în lucrările [5,8,19], îmbunătățind algoritmii și realizând mai multe experimente, fiind important să existe un generator de teste de diferite dificultăți în testarea unui sistem.

1.4 Studiul proprietății de accesibilitate a unei clase de sisteme parametrizate folosind “regular model checking”.

Lucrarea [4] studiază proprietatea de accesibilitate (reachability) a unei clase de sisteme parametrizate folosind o metodă numită “regular model checking”. Modulele fiecărui sistem sunt instanțiate dintr-un șablon sincronizat global și fiecare șablon sincronizat global este reprezentat de un automat cu număr finit de stări care are eveniment de tip global, dar și local. În articol a fost arătat că relațiile de accesibilitate sunt limbaje de tip stea (star languages), inchise la iterație. În sens invers, pentru orice asemenea limbaj există un șablon care conține numai evenimente globale care generează acel limbaj. Aplicații ale analizei proprietății de accesibilitate se găsesc de exemplu în analiza funcțiilor de control pentru sisteme distribuite care nu se blochează (deadlock-free). În particular, a fost arătat că funcții de control care au permisivitate maximă pot fi modelate cu automate cu număr finit de stări.

Rezultatele anterioare sunt foarte relevante pentru modelarea și analiza sistemelor cyber-physical deoarece oferă atât un model, cât și o metodă de analiză a existenței accesibilității (reachability), în

sensul că verifică automat că anumite stări pot fi atinse. În particular, este studiată clasa unor sisteme parametrizate. Un exemplu modern de asemenea sisteme parametrizabile care se sincronizează prin acțiuni comune este o mulțime de roboți de același tip care comunică între ei pentru a realiza un scop comun.

1.5 Testare Model-in-the-Loop pentru Sisteme Cyber Physical.

În zilele noastre, există un mare interes de a utiliza testarea automată, nu numai pentru că optimizează testarea manuală prin reducerea timpului și costului necesar, dar și pentru faptul că elimină erorile testarii manuală. Creșterea siguranței software-ului pentru controlul sistemelor complexe, care utilizează multe circuite electronice distribuite, necesită o testare amplă. În testarea bazată pe model, testele sunt derivate din cerințele sistemului și dintr-un model care descrie anumite aspecte funcționale și nefuncționale ale sistemului testat.

Primul pas al procesului de dezvoltare este proiectarea funcționalității sistemului. Aceasta înseamnă că trebuie să ne asigurăm că specificația trimisă dezvoltatorilor de software trebuie să fie corect proiectată. Avem nevoie să testăm sistemul la nivel de model. O optimizare a procesului de testare manuală poate fi testarea automată bazată pe model, folosind simulări cu modelul în buclă (model in the loop) și abordări bazate pe căutare.

După ce am investigat câteva abordări similare pentru testarea sistemelor Cyber Physical, am propus generarea de suite de testare la nivelul modelului în buclă, utilizând un algoritm genetic cu obiectiv multiplu [11,12,13]. Am identificat un set de cerințe pentru comportamentul produsului dorit și am căutat cazuri de testare care încalcă cerințele [14]. Algoritmul nostru de căutare se bazează pe o funcție obiectiv, creată prin formalizarea cerințelor controller-ului.

Abordarea noastră a fost ilustrată pe un sistem de control al vitezei de croazieră pentru o bicicletă cu propulsie hibridă, generând cazuri care pot fi utilizate în continuare pentru a testa sistemul la nivel de software în buclă și hardware în buclă.

1.6 Modelare și verificare formală folsind Event-B și platforma Rodin

Modelarea formală este esențială pentru definirea precisă, înțelegerea și raționamentul în proiectarea sistemelor complexe, cum ar fi sistemele cyberphysical. În cercetarea noastră am folosit limbajul Event-

B, o abordare formală pentru specificarea și verificarea sistemelor fiabile, susținută de platforma Rodin, bazată pe theorem proving, permițând un proces de specificare treptată pe baza rafinării. Folosim, de asemenea, din aceeași platformă, model checker-ul ProB și plug-in-ul iUML pentru vizualizarea

modelelor. Un studiu de caz al unui sistem de control al vitezei de croazieră pentru o bicicletă cu propulsie hibridă și pentru o bicicletă electrică (e-Bike) este prezentat în [15]. Abordarea noastră arată beneficiile utilizării unei platforme formale de modelare, care oferă multiple modalități de analiză a sistemului, în contextul sistemelor cyberphysical.

2. Colaborări internaţionale

Proiectul prilejuieşte întărirea legăturilor de colaborare cu grupuri puternice de cercetare din universităţi de prestigiu: Automotive Research Centre de la Universitatea din Bradford (cu colaborări importante cu Land Rover și Jaguar) si grupul Research Group on Natural Computing de la Universitatea din Sevilla, (dezvoltatorii celui mai avansat mediu de specificare şi simulare a P sistemelor, P-Lingua, avându-se în vedere integrarea metodelor propuse de noi pentru verificarea şi testarea sistemelor P /kP în acest mediu). Printre colaboratorii proiectului se numără Gexiang Zhang de la Universitatea Southwest Jiaotong din Chengdu. Se remarcă şi obţinerea recentă de către către prof. Gexiang Zhang a finanţării pentru un proiect de cercetare în domeniul calculului membranar, finanţat de National Science Fundation, China, în care sunt implicaţi doi dintre membrii echipei MACPS, Marian Gheorghe şi Florentin Ipate.

3. Formarea tinerilor cercetători

În proiect sunt implicaţi 3 tineri cercetători: un doctorand (Ana Țurlea, supervizată de Florentin Ipate) şi 2 cercetători post-doctorali (Sorina Preduț și Raluca Lefticaru). Ana Țurlea și Sorina Preduț sunt membri în echipa proiectului (fiind remuneraţi din fondurile acestuia), Raluca Lefticaru având statutul de colaborator. Se remarcă şi faptul că aceștia sunt co-autori la articolele [1, 3, 5, 7-20].

4. Publicaţii rezultate

În urma activităţii de cercetare din proiect, până în septembrie 2019 au rezultat următoarele lucrări:

- 4 articole [4,6,7,1] în jurnale cotate ISI (cu FI 5.451, 5.481, 0.83, 0.772, primele două de categoria A* conform standardelor naționale definite de către Comisia de Informatică din cadrul CNATDCU).
- 13 articole [3, 5, 8-11, 13-16, 18-20] în conferinţe internaţionale cu ISI proceedings.

- 1 articol acceptat în jurnal cotat ISI.
- 2 articole [2, 12, 17] trimise spre recenzie la jurnale cotat ISI.

Lista completă a lucrărilor:

1. M. Gheorghe, R. Ceterchi, F. Ipate, S. Konur, R. Lefticaru. Kernel P Systems. From Modelling to Verification and Testing. Theoretical Computer Science, 724, 45-60, 2018. IF 0.772.
2. F. Ipate, M. Gheorghe. Learning X-machines from queries and counterexamples, submitted.
3. R. Lefticaru, M. E. Bakir, S. Konur, M. Gheorghe, M. Stannett, F. Ipate. An Integrated Model Checking Toolset for Kernel P Systems. Int. Conf. on Membrane Computing 2017, LNCS, 183-195, 2017.
4. L. Lin, A. Stefanescu, W. Wang, R. Su, W.M. Wonham. Symbolic Reachability Analysis and Maximally Permissive Entrance Control for Globally Synchronized Templates. Automatica 87, pp. 290-300, 2018. ISI-indexed journal, IF 5.451.
5. A. Turlea, F. Ipate, R. Lefticaru: A Test Suite Generation Approach based on EFSMs using a Multi-Objective Genetic Algorithm. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2017), 153-160.
6. M E Bakir, S Konur, M Gheorghe, N Krasnogor, M Stannett. Automatic selection of verification tools for efficient analysis of biochemical models. Bioinformatics 34(18), 3187-3185, 2018; IF=5.481
7. E Csuhaj-Varju, M Gheorghe, R Lefticaru. P colonies and kernel P systems. International Journal of Advances in Engineering Sciences and Applied Mathematics 10(3), 181-192, 2018; IF=0.83
8. A. Ţurlea, F. Ipate and R. Lefticaru. Generating Complex Paths for Testing from an EFSM. 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, 242-249, 2018.
9. A. Turlea, F. Campean, R. Lefticaru. Search based Model in the Loop Testing for Cyber Physical Systems. Working Formal Methods Symposium, 2018.
10. M. Gheorghe, F. Ipate, R. Lefticaru, A. Ţurlea. Testing Identifiable Kernel P Systems using an X-machine Approach. Proceedings of the 19th International Conference on Membrane Computing (CMC19), 2018. **Best Student Paper Award**
11. A. Turlea. Test Suite Generation for Cyber Physical Systems at the Model in the Loop level. SYNASC, 2018.
12. A. Turlea, F. Campean. Model-in-the-Loop Testing for Cyber Physical Systems. Software Engineering Notes, submitted.
13. A. Turlea. Search based Model in the Loop Testing for Cyber Physical Systems. 2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), Bucharest, 2018.
14. F. Campean, U. Yildirim, E. Henshall. Synthesis of functional models from uses cases using the system state flow diagram: a nested approach. International Design Conference - Design 2018, 2833-2844, 2018.
15. S. Predut, F. Ipate, M. Gheorghe, F. Campean. Formal Modelling of Cruise Control System Using Event-B and Rodin Platform. 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th Intl. Conference on Data Science and Systems, 1543-1548, 2018.
16. A. Turlea, M. Gheorghe, F. Ipate. Search Based Software Engineering in Membrane Computing. 17th Brainstorming Week on Membrane Computing, Seville, 2019
17. A. Turlea, M. Gheorghe, F. Ipate, S. Konur. Search Based Software Engineering in Membrane Computing. 20th Conference on Membrane Computing, 2019
18. A. Turlea, M. Gheorghe, F. Ipate, S. Konur. Search Based Software Engineering in Membrane Computing. Journal of Membrane Computing, accepted.
19. A. Turlea. Testing Extended Finite State Machines using NSGA-III. In Proceedings of the 10th ACM

SIGSOFT International Workshop on Automating TEST Case Design, Selection, and Evaluation (A-TEST 19), August 26–27, 2019, Tallinn, Estonia.
20. A. Turlea. Search-based Testing using EFSMs. 30th International Symposium on Software Reliability Engineering (ISSRE 2019), Doctoral Symposium.

Director proiect,

Florentin Ipate