

Descrierea initiala a proiectului (februarie 2010):

TESTAREA SI MODELAREA FORMALA A APLICATIILOR SOFTWARE BAZATE PE SERVICII

Director de Proiect: ALIN STEFANESCU

Rezumat:

Testarea aplicatiilor software este o sarcina foarte costisitoare, dificila, care consuma mult timp si se estimeaza ca poate solicita pana la 50% din costul total al dezvoltarii sistemului software. Acest cost ar putea fi redus daca procesul testarii este cat mai mult automatizat. De aceea, sisteme care genereaza automat seturi de test sunt studiate de multi cercetatori si profesionisti. Una dintre cele mai atractive si mai recente abordari pentru automatizarea testelor este testarea bazata pe modele (MBT – model-based testing), care foloseste modele formale ce descriu sistemul de testat pentru generarea automata de teste. In timp ce teoria MBT a fost in principal dezvoltata si aplicata in ingineria software, o noua paradigma, aceea a arhitecturilor orientate pe servicii (SOA – service-oriented architectures), a castigat o rapida si extinsa popularitate inclusiv in lumea industriei, fiind identificata si de EU FP7 ca unul dintre domeniile cu potential inovativ la cel mai inalt nivel, contribuind la viziunea internetului de servicii (Internet of Services). Avantajele operabilitatii, flexibilitatii, adaptabilitatii si functionalitatii promise de SOA vin impreuna cu cateva provocari de rezolvat referitoare la robustețe, stabilitate si calitate. Scopul acestui proiect este de a propune noi solutii pentru asigurarea calitatii sistemelor bazate pe servicii folosind tehnologiile MBT. Proiectul va studia, pe de o parte, diferite tipuri de modele ce pot fi folosite in domeniul testarii integrarii serviciilor, si pe de alta parte, va adresa dificila problema a generarii automate de teste folosind o combinatie originala de tehnici MBT cu algoritmi evolutivi inspirati din biologie si algoritmi meta-euristici de cautare. Nu in ultimul rand, proiectul va ataca problema scalabilitatii cu care se confrunta de obicei inginerii, pastrand un strans contact cu furnizorii de software SOA, ca de ex. SAP, si validand rezultatele cercetarii pe benchmark-uri industriale.

Scopul principal urmarit

Acest proiect de cercetare va investiga metode noi si eficiente pentru testarea bazata pe modele a aplicatiilor software bazate pe servicii folosind tehnici evolutive. Ingredientele principale se afla in urmatoarele trei domenii: (a) modelare formala pentru testare pentru arhitecturi orientate pe servicii, (b) generare automata de teste si date de test si (c) aplicarea rezultatelor cercetarii pe studii de caz din practica. Scopul proiectului este acela de a defini metode nu numai avansate din punct de vedere teoretic, dar si aplicabile pe studii de caz reale intalnite in industrie.

Importanta si relevanta continutului stiintific

Arhitecturile orientate pe servicii (SOA – service-oriented architectures) sunt abordari recente din domeniul sistemelor distribuite in care serviciile software sunt entitatile de baza in structura si interoperabilitatea sistemului. SOA capata teren devenind predominante in industria software, dupa cum sunt prezentate intr-un studiu amplu efectuat de Forrester [1]. Acesta sustine ca 67% dintre companiile participante la studiu (mai precis, 2200 de companii din America de Nord si Europa) vor folosi SOA pana la sfarsitul anului 2009, in timp ce 60% dintre actualii utilizatori vor extinde intrebuintarea lor. Acest trend va continua cu siguranta conducand la realizarea Internetului de Servicii (**IoS – Internet of Services**), in care serviciile vor fi total disponibile in Internet. Sistemele bazate pe servicii (**SBS – service-based system**) vor prezenta avantaje semnificative in termeni de operabilitate, flexibilitate, adaptabilitate si functionalitate. Pe de alta parte, datorita caracterului distribuit si a lipsei de control asupra serviciilor, definirea unor asemenea SBS-uri necesita adresarea unor noi teme de cercetare in domeniul asigurarii stabilitatii si calitatii lor, in special in domeniul testarii SOA - v. [2,3,6,7].

Testarea bazata pe modele (MBT – model-based testing) este o clasa de tehnici de generare automata de teste pornind de la un model formal, ce descrie sistemul de testat, crescand astfel procentul de acoperire si eficienta procesului de testare. Paradigma MBT a primit in ultima perioada multa atentie atat in comunitatea academica, cat si industriala, fiind publicate cateva carti [11,12,13] si fiind organizate cateva workshop-uri dedicate domeniului (A-MOST, MBT, MOTES, MOTIP, etc.). In plus, exista deja o piata de produse comerciale MBT (precum Qtronic de la Conformiq, TestDesigner de la Smartesting, SpecExplorer de la Microsoft). Pe de alta parte, cele mai multe produse si metode sunt pentru testarea functionala in faza de dezvoltare software si doar cateva dintre ele sunt adaptate pentru paradigma SOA. Mai precis, putem identifica patru nivele diferite ale testarii SOA: *unit testing* (testarea operatiilor si a claselor interne), testarea singulara a serviciilor (testarea integrarii modulelor functionale intr-un serviciu singular), testarea integrarii serviciilor si testarea sistemului (ca intreg). Pana acum cercetarea s-a axat mai degraba pe primele doua tipuri de testari in detrimentul ultimelor doua, astfel ca imbunatatirea automatizarii testelor predomina activitatea dezvoltatorilor si furnizorilor de servicii in detrimentul consumatorilor sau integratorilor de servicii.

Dupa introducerea succinta a contextelor SOA si MBT, sa intelegem acum domeniul si sa accentuam problemele ramase deschise atat in cercetare cat si in

industrie. Astfel de provocari pot fi identificate in fiecare din pasii principali ai aplicarii MBT la sistemele bazate pe servicii. Acesti pasi sunt: (a) definirea si furnizarea modelelor formale pentru testare, (b) definirea metodelor pentru generarea automata de teste si date de test bazate pe aceste modele, si (c) aplicarea sau concretizarea testelor generate la sistemele bazate pe servicii. Mai jos sunt descrisi cei trei pasi separat:

(a) Modelarea formala pentru testarea SOA

Pana in momentul actual au fost propuse mai multe tehnici pentru asigurarea calitatii SOA (precum testare, monitorizare, verificare formala), examinate in [6,8]. Cu toate acestea, multe dintre aceste tehnici se bazeaza pe **limbaje si modele ad-hoc** pentru clase specifice de proprietati propuse spre testare si pentru produse specifice: de exemplu, cele 30 de abordari pentru testare SOA analizate in [8] folosesc **16 modele si limbaje diferite**. Acest lucru arata ca inca nu s-a ajuns la un consens si un standard pentru testarea SOA. Un motiv este faptul ca, pe de o parte modelarea SOA trebuie sa acopere mai multe nivele, de la descrieri de servicii singulare la compozitii de servicii complexe, iar pe de alta parte, cercetarea in domeniul MBT pentru SOA se afla inca intr-un proces de consolidare, influentat continuu de succesiunea rapida din dezvoltarea ingineriei serviciilor. Putem insa sa identificam doua mari grupuri de limbaje de modelare pentru SOA: bazate pe UML sau pe WS-*. Standardul UML (vezi <http://www.uml.org>) este de fapt limbajul de modelare pentru ingineria software clasica si exista cateva profile pentru UML care permit modelarea pentru testare, precum standardul U2TP (vezi <http://utp.omg.org>), si de asemenea profile UML recente pentru profile SOA (ca SoaML si UML4SOA). Totusi, **pana in prezent nu s-a cercetat combinarea celor doua profile care ar ingadui o abordare bazata pe MBT pentru SOA bazata pe profile UML**. Mai mult decat atat, in multe cazuri semantica UML nu este sufficient de precis definita pentru a fi folosit in MBT. Pe de alta parte, standardele de ingineria serviciilor sunt bazate pe limbaje de Servicii Web (**WS – Web Services**) (vezi <http://www.w3.org/standards>), care furnizeaza diferite tipuri de limbaje precum WSDL, WS-BPEL, WS-CDL pentru descrierea serviciilor. Desi cercetarea MBT ia in considerare modele ca WSDL si WS-BPEL pentru descrierea serviciilor singulare sau orchestratii de servicii, **nu s-a cerceteaza inca suficient domeniul testarii pentru coreografiilor de servicii** (diferite de orchestratii, coreografiile presupun un control distribuit fata de unul centralizat al comunicarii de servicii). Astfel de coreografii ar putea furniza o baza foarte buna pentru testarea integrarii de servicii [15]. Mai mult, noi standarde precum BPMN 2.0 ce permit modelarea de coreografii, nu au fost inca investigate din punct de vedere al MBT. Nu in ultimul rand, inca nu exista multe modele pentru testarea la nivel de sistem (system testing) a SBS. O astfel de testare necesita un lucru la nivel de interfata grafica (GUI – *Graphical User Interface*) pentru compozitii de servicii, si MBT pentru GUI reprezinta inca un domeniu nou de cercetare [10], in special pentru aplicatiile bazate pe servicii [16].

(b) Generarea si executia de teste pentru sistemele SOA

Fiind dat un model formal pentru testare, urmatorul pas MBT este acela de a genera automat teste (ca pentru a fi executate in sistemul de testat) in conformitate cu anumite criterii de acoperire. Metode foarte puternice de generare a testelor, in special pentru modele bazate pe stari, au fost elaborate in literatura [11]. Dar, in incercarea de a le aplica la sistemele bazate pe servicii, putem identifica cel putin trei dezavantaje importante:

- **Date de test:** Datorita distributivitatii, eterogenitatii si a cuplarii slabe (*loose coupling*), sistemele bazate pe servicii se bazeaza in cele mai multe cazuri pe comunicarea asincrona folosind date complexe de tip XML. Aceste date expun dependente interne complexe precum si referinte la datele existente in sistem (*master data*) [17]. Astfel, **furnizarea datelor de test pentru asemenea sisteme devine o activitate intensa pentru testerii si mareste gradul de dificultate pentru generarea automata de date de test.** Desi exista multe metode de generare a datelor de test in literatura [19], bazate de exemplu pe: solutionarea constrangerilor (*constraint solving*), executie simbolica (*symbolic execution*), reducerea domeniului (*domain reduction*), sau cautare meta-euristica (*metaheuristic search*), aceste metode au fost in special focusate pe testarea cutie-alba (*white-box testing*), adica persoana care testeaza are acces la codul aplicatiei testate. Dar de cele mai multe ori aceasta nu este posibil pentru aplicatiile bazate pe servicii, in care numai interfetele de servicii si protocoalele de comunicare sunt expuse.
- **Testarea sistemului si a integrarii:** In al doilea rand, cele mai multe abordari in literatura de specialitate [6] se concentreaza mai degraba pe testarea serviciului singular, decat pe testarea integrarii si a sistemului ca intreg. Lucrand sub ipoteza de integrare stransa a serviciilor (*hard-coded integration*), ar putea fi aplicabile tehnici existente pentru testarea integrarii sistemelor bazate pe componente. In principal, acestea folosesc modele comportamentale si statice precum UML (diagrame de clase, masini cu stari, diagrame de colaborare si interactiuni) pentru a genera teste. Cu toate acestea, ipoteza de integrare stransa nu este valabila in domeniul IoS unde compozitiile de servicii se cupleaza slab. Mai mult decat atat, multe dintre modelele SOA utilizate in practica provin mai degraba din domeniul standardelor WS decat din cele UML si informatiile necesare pentru generarea de teste lipsesc din meta-modele.
- **Design vs. testare in timp real:** Nun in ultimul rand, avand in vedere natura dinamica si adaptabila a SBS-urilor, multe dintre tehnicile traditionale MBT care implica numai generarea de teste la momentul design-ului ar putea sa nu fie foarte eficiente in depistarea problemelor legate de integrare, care apar in timp real la executie (*at runtime*) sau in timpul reconfigurarii dinamice a sistemului, dupa cum sunt prezentate in [7]. Prin urmare, **exista o nevoie clara de noi tehnici MBT adaptate pentru testarea in timp real (*runtime testing*)**, vizand in mod explicit erorile din timpul operatiunilor de publicare de servicii (*service publishing*), descoperire de servicii (*service discovery*), compozitie de servicii (*service composition*) si atasare de servicii (*service binding*) in timp real.

(c) Aplicarea in industrie a metodelor de cercetare MBT

In paralel cu dificultatile din punct de vedere stiintific identificate mai sus, exista si o nevoie stringenta de imbunatatire a testarii in industrie. Concluziile raportului Gartner [9] arata ca practica e departe de a face fata cerintelor IoS din punctul de vedere al calitatii, deoarece solutiile software de asigurare a calitatii in domeniul SOA (de ex. Green Hat, Parasoft, Soasta, Progress Software, iTKO) **se concentreaza mai degraba pe testarea si monitorizarea serviciilor singulare decat pe compozitii complexe de servicii, si mai mult pe arhitecturi statice decat pe cele dinamice.** Un motiv este faptul ca informatiile (precum descrierea formala a compozitiilor de servicii si a datelor complexe de test) care ar putea fi folosite la automatizarea acestui process lipsesc, atat in timpul conceptiei cat si al executiei sistemului. Acest fapt este confirmat si de interviurile realizate in industrie [4,5] care arata ca **majoritatea proiectelor SOA si software sufera de o testare inadecvata si insuficienta.** Un alt motiv pentru aceasta situatie problematica este lipsa de tehnici complete si consistente, fortandu-i pe multi manageri IT sa implementeze procese de testare ad-hoc si *in-house*. In plus, exista o proliferare a limbajelor dedicate (**DSL – domain-specific language**) utilizate in industrie care sunt foarte benefice in contextul dat datorita inaltei specializari, dar au dezavantajul clar ca **tehnicele MBT nu sunt usor de aplicat din cauza diverselor particularitati ale DSL-urilor** ce trebuie sa fie luate in considerare. De exemplu, SAP, liderul pe piata aplicatiilor pentru intreprinderi (*business applications*) si un promotor al Enterprise SOA [18], foloseste mai mult de 13 tipuri de DSL-uri pentru arhitectura sa SOA. Nu in ultimul rand, asa cum se mentioneaza in [11], concretizarea in practica a testelor abstracte generate folosind MBT (adica facandu-le executabile prin adaugarea de date de test manuale sau setarea configuratiilor testelor) poate presupune acelasi efort ca si pentru pasii de modelare si generare de teste.

Testarea evolutiva (ET - Evolutionary Testing) este un caz particular de abordare bazata pe cautare pentru generarea datelor de test. Tehnicile bazate pe cautare (precum algoritmi genetici si algoritmi evolutivi) sunt concepute spre a gasi aproximari bune pentru solutii optime in spatii complexe de cautare, in special in cazurile dificile cand metodele de generare a testelor clasice si formale esueaza datorita exploziei combinatoriale a datelor de test. Cautarea este determinata de asa-numitele functii de fitness, care sunt utilizate pentru a optimiza datele de test spre gasirea unei erori in comportamentul sistemului. Tehnicile evolute aplica algoritmi inspirati din biologie, cum ar fi reproducerea, mutatia, recombinarea, selectia naturala si supravietuirea celui mai adaptat, pentru a gasi date de test adecvate pentru optimizarea functiilor de fitness. Pana in prezent, testarea evolutiva s-a concentrat in aplicarea acestor tehnici in testarea (structurala) cutie-alba (*white-box testing*), asa cum se vede din studiul de la [20]. Aplicarea tehnicilor bazate pe cautare in domeniul testarii cutie-neagra (*black-box testing*) sau a testarii bazate pe modele, este un subiect mult mai putin cercetat [21]. Desi fiind o tehnica promitatoare care a atras, de asemenea, finantarea de catre UE in proiectul EvoTest (EvoTest - Evolutionary Testing for Complex Systems - 2006-2009), **aplicarea testarii evolute la testarea SOA nu a fost inca luata in considerare.** Am putut identifica doar cateva lucrari foarte recente care utilizeaza tehnici bazate pe cautare si chiar si

acolo accentul s-a pus pe *service level agreement*-uri [22] si orchestratii [23]. Astfel, acesta este un domeniu larg neexplorat pe care il vom investiga in proiectul nostru.

Interdisciplinaritatea proiectului propus

Acest proiect va asigura **o legatura puternica intre doua domenii distincte din informatica: informatica teoretica** (enumerata ca domeniu prioritar 1.5 de catre CNCNIS-UEFISCU sub Stiintele Naturii - a se vedea anexa 2) si **ingineria calculatoarelor si ingineria sistemelor** (enumerata ca domenii prioritare 2.2 de catre CNCNIS-UEFISCU sub Stiintele Ingineresti - a se vedea anexa 2).

Modelarea formala si testarea bazata pe modele apartin primului domeniu, in timp ce sistemele bazate pe servicii apartin celui de-al doilea, sinergiile care exista intre cele doua domenii fiind doar foarte recent studiate. In plus, rezultatele obtinute vor fi aplicate in **studii de caz multidisciplinare**, efectuate in colaborare cu cercetatorii de la SAP, privind modelarea si testarea serviciilor enterprise. Prin studii de caz multidisciplinare intelegem de pilda scenariu din aria Internetului Lucrurilor (*Internet of Things* – tema prioritara pentru EU, vezi [25]), unde retele dinamice de obiecte (telefoane mobile, masini, aparate casnice inteligente) comunica folosind tehnologii SOA. In cele din urma, la un nivel tehnic, exista **o alta interdisciplinaritate ridicata data de utilizarea algoritmilor evolutivi (inspirati din biologie), precum si aplicarea lor la testarea SOA**, asa cum va fi descris in sectiunea urmatoare.

Bibliografie:

- [1] Forrester Research, "Enterprise and SMB software survey", North America and Europe, Q4 2008. Survey, 2008.
- [2] Aberdeen Group, "SOA and Web Services Testing – How Different Can It Be?" Industry Survey. August 2007.
- [3] D. Linthicum, and J. Murphy, "Key Strategies for SOA Testing". Mindreef Inc., 2007.
- [4] PR NEWSWIRE, "More than two-third (68%) of European IT Managers Believe That Products and Systems are Launched With Insufficient Testing," Sep. 29, 2005. [Online]. Available: <http://www.ulitzer.com/node/135839>
- [5] Green Hat, "Nine out of ten survey respondents agreed that inadequate testing is a major issue in delivering successful SOA projects using messaging middleware," [Online]. Available: <http://www.greenhat.com/news/newsTYIUSSSXK.html>
- [6] G. Canfora, M. Di Penta, "Service-oriented architectures testing: A survey," *Software Engineering*, Springer, pp. 78-105, 2009.
- [7] M. Greiler, H.-G. Gross, K. A. Nasr, "Runtime Integration and Testing for Highly Dynamic Service Oriented ICT Solutions – An industry challenges report," in TAICPART'09, IEEE Computer Society, 2009, pp. 51-55.
- [8] S-Cube - the European Network of Excellence in Software Services and Systems. "Survey of Quality Related Aspects Relevant for Service-based Applications". Deliverable PO-JRA-1.3.1, 2008. [Online]. Available: <http://www.s-cube-network.eu/results/deliverables/wp-jra-1.3/PO-JRA-1.3.1-Survey-of-quality-related-aspects-relevant-for-SBAs.pdf>
- [9] Gartner, "Hype Cycle for Application Development, 2009". Gartner Research, July 2009.

- [10] A. M. Memon, "An event-flow model of GUI-based applications for testing," *Softw. Test., Verif. Reliab.*, vol. 17, no. 3, pp. 137–157, 2007.
- [11] M. Utting, B. Legeard, "Practical model-based testing, a tools approach". Morgan Kaufmann, 2007.
- [12] P. Baker, Z. R. Dai, J. J. Grabowski, Ø. Haugen, I. Schieferdecker, and C. Williams, "Model-Driven Testing: Using the UML Testing Profile". Springer, 2008.
- [13] J. Jacky, M. Veanes, C. Campbell, W. Schulte, *Model-based Software Testing and Analysis with C#*. Cambridge University Press, 2008.
- [14] M. Di Penta, G. Canfora, G. Esposito, V. Mazza and M. Bruno, "Search-based Testing of Service Level Agreements." In Proc. of the Conf. on Genetic and Evolutionary Computation (GECCO'07), pp. 1090–1097. ACM Press, 2007.
- [15] A. Stefanescu, S. Wiczorek, A. Kirshin. "MBT4Chor: a model-based testing approach for service choreographies". In Proc. of 5th European Conf. on Model-Driven Architecture Foundations and Applications (ECMDA'09), LNCS, volume 5562, pp. 313-324. Springer, 2009.
- [16] S. Wiczorek, A. Stefanescu. "Improving Testing of Enterprise Systems by Model-based Testing on Graphical User Interfaces". To appear in the Proc. of the 7th Workshop on System Testing and Validation (STV'10), IEEE Computer Society, 2010.
- [17] S. Wiczorek, A. Stefanescu, I. Schieferdecker. "Test data provision for ERP systems." In Proc. of the International Conference on Software Testing, Verification, and Validation (ICST'08), pp. 396-403. IEEE Computer Society, 2008.
- [18] D. Woods, T. Mattern, "Enterprise SOA – Designing IT for Business Innovation". O'Reilly, 2006.
- [19] B. Korel, "Automated Software Test Data Generation." *IEEE Trans. Software Eng.* 16(8): 870-879, 1990.
- [20] P. McMin, "Search-based software test data generation: a survey." *Softw. Test., Verif. Reliab.*, 14(2):105–156. Wiley, 2004.
- [21] Raluca Lefticaru, Florentin Ipate, "Automatic State-Based Test Generation Using Genetic Algorithms". Proc. of SYNASC'07, pp. 188-195. IEEE Press, 2007.
- [22] M. Di Penta, G. Canfora, G. Esposito, V. Mazza, M. Bruno, "Search-based testing of service level agreements". In Proc. of GECCO'07, pp. 1090–1097, ACM Press, 2007.
- [23] R. Blanco, J. García-Fanjul and J. Tuya, "A first approach to test case generation for BPEL compositions of web services using Scatter Search". In Proc. of 2nd Int. Workshop on Search-Based Software Testing, pp. 131-140, IEEE Computer Society, 2009.
- [24] Commission of the European Communities. "Internet of Things — An action plan for Europe". June 2009. Online at: http://ec.europa.eu/information_society/policy/rfid/documents/commiot2009.pdf

Obiectivele, metodologia si rezultatele preconizate ale cercetarii

Proiectul are ca obiectiv principal asigurarea unei **perspective integrate si complete (end-to-end) pentru MBT pentru integrarea de servicii**, incepand cu modele de testare bazate pe ultimele dezvoltari si modele din industrie, continuand cu algoritmi noi pentru MBT si ET optimizati pentru aceste modele si in final validarea fezabilitatii lor pe scenarii SOA din practica. Avand in vedere intervalul de timp de 24 de luni, proiectul propus nu va acoperi toate problemele deschise identificate in sectiunea anterioara, ci se va concentra pe o serie de teme cu un **impact ridicat in cercetare, impact bazat pe experienta ampla anterioara a aplicantului in MBT pentru SOA** [1,2,3,4,5]. Obiectivele detaliate si ideile de rezolvare sunt grupate in functie de cele trei domenii identificate in sectiunea precedenta:

(a) Modelare formala pentru testarea SOA

▪ **Obiectivul a.1) Imbunatatirea testabilitatii pentru modelele SOA.**

Cele mai multe dintre lucrarile existente in MBT se bazeaza pe UML clasic sau pe alte modele bazate pe stari, inasa este inca nevoie de cercetare in domeniu pentru adaptarea acestor tehnici la profilele UML pentru SOA (UML4SOA sau SoaML) si combinarea lor cu profile UML existente pentru testare (U2TP). Dupa cum s-a mentionat in sectiunea anterioara, nu suntem la curent cu existenta vreunei lucrari care sa combine cele doua profile intr-un cadru integrat. Aplicantul a inceput foarte recent sa investigheze aplicabilitatea profilului U2TP la integrarea de servicii bazate pe coreografii - a se vedea [1]. In acest proiect, vom combina aceasta cercetare cu profiluri UML recent propuse pentru SOA. In plus, aplicantul a efectuat in [2] investigatii cu privire la informatiile necesare pe care modelele pentru coreografii trebuie sa le contina pentru o abordare MBT eficienta. Vom folosi aceste cunostinte pentru verificarea si cresterea testabilitatii modelelor coreografice existente, cum ar fi WS-CDL sau BPEL4Chor. Acest lucru va permite o utilizare mai intensa a MBT pentru testarea coreografiilor, completand astfel cercetarea actuala axata doar pe studiul orchestratiilor de servicii.

▪ **Obiectivul a.2) Imbunatatirea modelarii pentru testarea GUI pentru SBSs.** De asemenea, aplicantul a investigat recent testarea la nivel de interfata grafica (GUI) pentru aplicatiile bazate pe servicii [3]. Cum interesul in acest domeniu este foarte mare atat in industrie cat si in cercetare, acest obiectiv va investiga in special modele bazate pe stari pentru GUI, re folosind elemente din meta-modele de testare.

(b) Generarea de teste pentru sisteme SOA folosind tehnici evolutive

Cum algoritmi genetici si evolutivi implica, prin natura lor, adaptabilitate si evolutie, ei sunt o alegere buna pentru contextul dinamic al IoS, unde modificari si reconfigurari, presupunand consistenta sistemului, se intampla in timp real. Cu toate acestea, datorita probabil progresului foarte recent si rapid al domeniului SOA, testarea evolutiva (ET) nu a fost inca utilizata in acest domeniu. Ca o abordare inovatoare, vom cerceta utilizarea ET in combinatie cu MBT pentru SOA, asa cum este descris mai jos:

▪ **Obiectivul b.1) Generarea datelor de test folosind tehnici evolutive.**

Aplicantul are o intelegere profunda a datelor de test pentru SOA si a problemelor aferente [4]. Furnizarea de date de test este dificila in domeniul SBS din cauza multiplelor constrangeri intre diferite campuri de intrari de date si constrangeri asociate proceselor si configuratiilor sistemului. Vom aplica algoritmi evolutivi si genetici pentru generarea diferitelor mutatii de date care au ca scop evolutiv satisfacerea constrangerilor aferente. Acest lucru ar trebui sa ofere o alternativa viabila la metodele de generare de teste, bazate fie pe satisfacerea constrangerilor, care are un grad de complexitate mare (NP-complet), fie pe algoritmi de verificare a modelelor (*model-checking*) care au problema exploziei de stari si spatiu. Mai mult, tehnicile actuale de generare a datelor de test sunt dezvoltate in principal pentru parametrii cu tipuri de date simple, ceea ce nu este cazul si pentru testarea SOA. ET este destul de versatil pentru a lucra pe structuri de date de tip arbore, care pot manipula astfel parametrii complecsi de intrare ceruti de servicii.

- **Obiectivul b.2) Investigarea de noi combinatii intre ET si MBT pentru SOA.** Aplicantul a publicat deja o metoda MBT pentru coreografii de servicii [5]. Aceasta a fost o prima abordare care aplica algoritmi de randomizare (implementati de catre un co-autor de la IBM) pentru cautarea in spatiul de stari al datelor de intrare de test. In acest obiectiv vom adapta tehnici ET existente care poata adresa noi modele SOA bazate pe stari care descriu interactiunile de servicii. In consecinta, vor fi dezvoltate noi tehnici bazate pe cautare pentru testarea de modele de comunicare, cum ar fi coreografiile de servicii, pentru a face fata problemelor de testare suplimentare care apar in faza de integrare a serviciilor. De notat ca ET pentru specificatii bazata pe stari este un domeniu recent de cercetare. Vom combina ET cu MBT, astfel incat criteriile de acoperire a diferitelor stari sau tranzitii pentru coreografii sunt realizate prin exercitarea de *triggere* si *garzi logice* (*guard*) corespunzatoare apelurilor de servicii. Tehnicile de cautare meta- euristice nu sunt algoritmi de sine statatori, ci mai degraba strategii adaptabile la probleme specifice. Vom rafina astfel de strategii la elementele specifice ale coreografiilor de servicii, cum ar fi semantica diferitelor mesaje de comunicare (expediere, primire sau semantici de observare) sau consistenta din punct de vedere local si global.
- **Obiectivul b.3) Studiul tranzitiei de la testarea statica la testarea in timp real.** Una dintre caracteristicile sistemelor bazate pe servicii este schimbarea centrului de interes de la static la timp real, cu problemele asociate. Noi clase de erori vizibile numai in timp real trebuie sa fie abordate precum atasarea tarzie de servicii (*late binding*), erori de compozitie, sau incalcari de *service agreement*-uri. In testarea in timp real, generarea si executia testelor sunt impletite. Avantajul testarii in timp real este ca generarea de teste poate folosi raspunsul actual al sistemului inainte de calcularea urmatorului pas de testare si ca poate beneficia de datele obtinute prin monitorizarea serviciilor. Avem in vedere cel putin doua moduri in care ET ar putea ajuta. Primul, este acela de a directiona in mod inteligent cautarea in bazele de date obtinute prin monitorizarea serviciilor, pentru a extrage date de testare adecvate (acest lucru este esential datorita dificultatii de a modela si a genera date de test "bune" in

timp real). In al doilea mod, avand in vedere conceptele sale native de adaptare si de mutatie, ET ar putea imbunatati algoritmi MBT, permitandu-le adaptarea si transformarea in timp real, pentru a reactiona in timp util si in mod constant la adaptarea sistemului de testat in timp real.

(c) Aplicarea in industrie a metodelor de cercetare

- **Obiectivul c.1) Definirea unui studiu de caz exploratoriu (*exploratory use case*) pentru o buna ajustare a metodelor de mai sus.** In scopul de a valida si efectua in mod corespunzator experimente asociate cu cercetarea de mai sus, vom defini un studiu caz exploratoriu (*exploratory use case*) [6]. Acest lucru va presupune: (1) crearea cadrului de testare implementand algoritmi MBT si ET si (2) definirea unui studiu de caz academic pentru un sistem bazat pe servicii, incluzand atat modele de testare, precum si o implementare a sistemelor bazate pe servicii. Studiul de caz concret va fi un scenariu simplificat precum *Purchase Order – Sales Order* (luat de exemplu, de la benchmark-uri open-source din Web Service Test Forum - <http://www.wstf.org/docs/scenarios>), impreuna cu o implementare open-source a scenariului folosind de exemplu cadrul Open Services Gateway Initiative (OSGi). Cadrul pentru testare va fi bazat pe platforma Eclipse, care ofera o bogata biblioteca de plug-in-uri de modelare si testare, si permite o integrare usoara a noilor implementari. Avand in vedere ca tehnicile evolutive sunt tehnici generice care trebuie sa fie reglate la problema in cauza (in cazul nostru domeniile MBT si testare SOA), ne propunem sa realizam atat o analiza teoretica a problemelor, precum si masuratori experimentale, in scopul de a gasi cele mai bune setari ale parametrilor de cautare.
- **Obiectivul c.2) Adaptarea metodelor dezvoltate la domeniul Enterprise SOA.** Una dintre cele mai mari probleme cu care se confrunta inginerii software si testerii este aceea a scalabilitatii rezultatelor, astfel ca multe abordari care sunt atractive si elegante in laborator se dovedesc a fi inaplicabile in practica. Proiectul va tine seama de la inceput de problema scalabilitatii, folosind simularea si experimentarea nu numai pentru studiul de caz academic de la obiectivul c.1, dar si pentru benchmark-uri din industria software. Obiectivul va fi Enterprise SOA. Mai concret, **aceasta va fi posibil datorita experientei vaste a applicantului dobandita in ultimii 3 ani in cursul cercetarilor sale postdoctorale in laboratorul SAP Research.** Applicantul cunoaste pana in cele mai mici detalii platforma SOA implementata la SAP si este constient de experienta de zi cu zi a testerilor de la SAP din zecile de ore petrecute in intalniri si discutii. Mai mult, diverse nevoi in testarea SOA aparute de la parteneri industriali din MODELPLEX sau DEPLOY (cum ar fi Telefonica, Thales, Siemens, Bosch) vor fi luate in considerare, in special scenariii din domeniul Internetului Lucrurilor. Deoarece o parte din modelare e facuta folosind DSL-uri (pentru coreografiile sau GUI), punerea in aplicare a rezultatelor cercetarii va fi realizata prin transformari de modele in limbaje generice ca UML.

Metodologia cercetarii

Obiectivele proiectului propus pot fi impartite in teoretice, experimentale si de simulare, care reprezinta trei arii metodologice distincte. In oricare dintre cazuri, metodologiile de cercetare vor urma indicatiile generale recomandate pentru informatica teoretica [7], ingineria software [8] si ingineria software empirica [9,6].

Obiectivele a.1-a.2 si b.1-b.3, de exemplu propunand formalisme de modelare de teste SOA adecvate pentru a fi codificate in algoritmi evolutivi, apartin informaticii teoretice. Metodele folosite in acest caz vor implica: crearea modelelor formale (UML sau WS-*), definirea conceptelor dintre ele, demonstrarea proprietatilor prin metode deductive. Rezultatele teoretice vor fi judecate dupa cunostintele pe care le dezvaluie despre natura matematica a acestor modele si dupa utilitatea lor in practica computationala si a usurintei de aplicare. Obiectivele proiectului apartinand ingineriei software vor consta in realizarea de experimente care vor urma bine-cunoscutele sabloane din [6,9].

Bibliografie:

- [1] A. Stefanescu, M.F. Wendland, S. Wiczorek. "Using the UML testing profile standard for testing of enterprise service choreographies." *Submitted* to the European Conference on modeling foundations and applications (ECMFA'10), Springer proceedings, 2010.
 - [2] S. Wiczorek, A. Roth, A. Stefanescu, A. Charfi. "Precise steps for choreography modeling for SOA validation and verification." In Proc. of International Symposium on Service-Oriented System Engineering (SOSE'08), pp.148-153. IEEE Computer Society, 2008.
 - [3] S. Wiczorek, A. Stefanescu. "Improving Testing of Enterprise Systems by Model-based Testing on Graphical User Interfaces." To appear in the Proc. of the 7th Workshop on System Testing and Validation (STV'10), IEEE Computer Society, 2010.
 - [4] S. Wiczorek, A. Stefanescu, I. Schieferdecker. "Test data provision for ERP systems." In Proc. of the International Conference on Software Testing, Verification, and Validation (ICST'08), pp. 396-403. IEEE Computer Society, 2008.
 - [5] A. Stefanescu, S. Wiczorek, A. Kirshin. "MBT4Chor: a model-based testing approach for service choreographies." In Proc. of 5th European Conference on Model-Driven Architecture Foundations and Applications (ECMDA'09), LNCS, volume 5562, pp. 313-324. Springer, 2009.
 - [6] R. K. Yin. "Case Study Research, Design and Methods." Sage Publications, 2nd edition, 1994.
 - [7] G. Dodig-Crnkovic. "Scientific Methods in Computer Science.", Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, 2002.
 - [8] C. B. Seaman. "Qualitative Methods in Empirical Studies of Software Engineering" IEEE Transactions on Software Engineering, Vol. 25, No. 4, pp.557-572, 1999.
- C. Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell, A. Wesslen.
"Experimentation in Software Engineering - An Introduction." Kluwer, 2000.

Gradul de originalitate/inovare si impactul preconizat al proiectului

Originalitatea proiectului

Proiectul propus va contribui la domeniul de cercetare foarte activ al modelarii si testarii SOA, unde exista multe probleme deschise dupa cum reiese din sectiunea 9.2. Exista noi standarde pentru SOA introduse de la an la an, iar trendul va continua cu siguranta. Toate aceste modele si standarde trebuie sa fie completate cu o perspectiva consistenta din punct de vedere al testarii (a se vedea Obiectivele a.1-a.2). Solutiile propuse se bazeaza pe doua domenii recente de succes, MBT si ET. Ele urmeaza sa fie combinate intr-un mod diferit de cel al aplicariilor lor. Nu suntem la curent cu nici o asemenea incercare in acest domeniu pana acum. In plus, cerintele speciale ale testarii SOA, inclusiv testarea in timp real, reprezinta un domeniu nou in care progrese clare se pot face cu ajutorul MBT si ET atat separat, cat si combinate (a se vedea Obiectivele b.1-b.3).

Complexitatea problemelor cercetate

Domeniul SOA adauga o noua dimensiune de complexitate la deja cunoscutele dificultati din ingineria software. Sistemele SOA sunt de obicei sisteme mari care acopera nu numai activitatile unei singure intreprinderi, ci si integrarea inter-organizationala depasind granitele intreprinderilor. De exemplu, Enterprise SOA dezvoltat la SAP folosind modele foloseste o duzina de tipuri diferite de modele care contin informatii despre obiectele business, unitati de *deployment*, componente de servicii, interfete de servicii, scenarii de integrare, variante de procese business, coreografii de servicii, etc.. Un asemenea sistem contine cateva mii de servicii, sute de coreografii si milioane de linii de cod implementate. Testarea unui astfel de sistem software imens implica mii de testeri folosind mai multe tipuri de testare. Este esentiala utilizarea de tehnici puternice de automatizare pentru a face fata complexitatii mari a sistemului. MBT este un candidat ideal pentru a utiliza informatiile abstracte din modele SOA pentru generarea de teste. ET adauga algoritmi inteligenti pentru a furniza date de test corecte pentru a face testele generate executabile. Cu toate acestea, pentru a face acest lucru in modul cel mai eficient e nevoie de cercetare inovatoare si multa expertiza. Proiectul propus are ca scop atacarea acestei probleme complexe si motivante.

Impactul preconizat

In primul rand, Comisia Europeana a evaluat arhitecturile orientate pe servicii si ingineria serviciilor ca domenii cu potential extrem de ridicat de inovare si impact asupra societatii informationale europene. O confirmare este bugetul mare de finantare pus la dispozitie in cadrul FP7-ICT Programme for Research and Development Objective 1.2 "Internet of Services, Software and Virtualisation" si "Service and Software Architectures, Infrastructures and Engineering". Ultimul FP7 call 5 va oferi 90 milioane de euro pentru urmatorul val de proiecte in acest Objective 1.2. Pe de o parte, **proiectul propus va contribui direct la viziunea europeana a "Internetului de Servicii"** si pe de alta parte va beneficia de ultimele progrese din domeniu in urmatorii 2 ani. Mai mult, testarea evolutiva a fost gasita de asemenea evaluata ca promitatoare de catre Comisia Europeana, ca dovada fiind finantarea a unui intreg proiect de cercetare pe aceasta tema: EvoTest (Evolutionary Testing for Complex Systems -

<http://evotest.iti.upv.es>). EvoTest nu a atins domeniul SOA, insa **proiectul nostru cauta noi aplicatii ale ET in domeniul SOA.**

In al doilea rand, **exista un interes in mod clar semnalat de industrie pe tema proiectului propus.** Dupa cum se stie, exista o diferenta semnificativa intre *inventie* si *inovatie*, inovatia fiind o suma de inventii aplicate cu succes in practica. (Pus in termeni metaforici de business: "inventie este conversia banilor in idei, in timp ce inovatia este transformarea ideilor in bani"). Pe baza acestui fapt, am orientat cercetarea noastra si spre aplicarea ei in practica. Astfel, asa cum am mentionat in cadrul Obiectivului c.2, vom tine legatura stransa cu cei din industrie. **Mai precis, aplicantul va continua sa colaboreze indeaproape cu colegii sai de la SAP Research si va aplica rezultatele cercetarii la exemple SAP.** In ultimii ani aplicantul a fost responsabil in interiorul SAP cu promovarea MBT pentru SOA. Subiectul va continua sa fie cercetat de SAP in urmatoorii ani, precum si de diferite proiecte: Proiectul FP7 DEPLOY se va derula pana in 2012 si unul din subiectele prioritare pentru SAP este MBT pentru SOA. In plus, SAP Research va participa intr-un proiect mare "Software cluster" (finantat de guvernul german, a se vedea <http://www.ideen-zuenden.de/de/468.php>), in care asigurarea calitatii SOA va fi printre subiectele investigate. Mai mult decat atat, European Institute of Innovation and Technology (EIT) tocmai a anuntat recent finantarea unui laborator EIT ICT pe tema *Future Internet* (vezi <http://www.eitictlabs.eu>). SAP, Institutul Fraunhofer si Universitatea Tehnica din Helsinki fac parte din consortiu si aplicantul a colaborat deja cu persoane implicate in acest proiect. Nu in ultimul rand, aplicantul a colaborat de asemenea cu grupul de cercetare de la IBM Research Haifa pe tema MBT si va continua acest lucru si in timpul proiectului. In concluzie, **toate activitatile proiectului vor fi pe deplin sincronizate cu aceste proiecte si vor contribui la impactul generat de acestea.**

Modul de organizare a proiectului (managementul proiectului)

An	Obiective	Activitati
AN I	Obiectivul (a): Modelarea formala pentru testarea SOA	Activitatea a.1) <i>Imbunatatirea testabilitatii pentru modelele SOA:</i> Cercetare si scriere de lucrari despre profile UML pentru testarea SOA. Colaborari cu IBM si TU Berlin
		Activitatea a.2) <i>Imbunatatirea modelarii pentru testarea GUI pentru SBSs:</i> Cercetare si lucrari despre modelarea GUI in colaborare cu SAP.
	Obiectivul (b): Generarea de teste pentru sisteme SOA folosind tehnici evolutive	Activitatea b.1) <i>Generarea datelor de test folosind tehnici evolutive:</i> Cercetare si scrierea unor lucrari despre generarea datelor de test, cu colegi de la Universitatea Pitesti

			Activitatea b.2) <i>Investigarea de noi combinatii intre ET si MBT pentru SOA</i> : Cercetare si publicatii. Stabilirea unei colaborari si vizite in grupul de testare din Sheffield
	3	Obiectivul (c): Aplicarea in industrie a metodelor de cercetare	Activitatea c.1) <i>Definirea unui studiu de caz exploratoriu (exploratory use case)</i> : cumparea unor echipamente hardware competitive si licente software si inceperea implementarii studiului de caz pentru experimentare
			Activitatea c.2) <i>Adaptarea metodelor dezvoltate la domeniul Enterprise SOA</i> : prima vizita la SAP
AN II	1	Obiectivul (a): Modelarea formala pentru testarea SOA	Finalizarea lucrarilor din Anul I si scrierea unui articol de jurnal cu toate contributiile
	2	Obiectivul (b): Generarea de teste pentru sisteme SOA folosind tehnici evolutive	Activitatea b.2) <i>Investigarea de noi combinatii intre ET si MBT pentru SOA</i> : scrierea unui articol de jurnal
			Activitatea b.3) <i>Studiul tranzitiei de la testarea statica la testarea in timp real</i> : cercetarea descrisa in Sectiunea 9.3
3	Obiectivul (c): Aplicarea in industrie a metodelor de cercetare	Activitatea c.2) <i>Adaptarea metodelor dezvoltate la domeniul Enterprise SOA</i> : cercetare si a doua vizita la SAP pentru aplicarea rezultatelor cercetarii	

Fezabilitatea proiectului (posibile riscuri, solutii alternative)

Fezabilitatea si credibilitatea proiectului sunt sustinute de urmasorii factori esentiali:

- aplicantul este un expert in subiectele propuse in proiect, cu experienta acumulata la universitati de prestigiu (München, Edinburgh, Konstanz, Bucuresti), precum si la laboratorul SAP Research, cu multe lucrari in domeniu publicate in ultimii ani.
- grupul puternic de cercetare de la Universitatea Pitesti, prof. Florentin Ipate, prof. Tudor Balanescu, si colaboratorii lor mai tineri. Avand in vedere numarul de publicatii, impactul si vizibilitatea, **grupul de testare software de la Pitesti este in prezent cel mai puternic din Romania**. Prof. Ipate coordoneaza un proiect in intervalul de timp 2008-2011, cu titlul "*O abordare evolutiva integrata de modelare formala si testare*", finantat de CNCIS, in cadrul PN-II-ID-PCE-2008-2. Acest lucru se potriveste foarte bine cu proiectul

nostru cu aplicatii SOA.

- Universitatea Pitesti are o relatie foarte stransa cu grupul "Testing and Verification" de la Universitatea Sheffield, condus de prof. Marian Gheorghe. Acest grup este lider in Europa in domeniul testarii, in special al testarii evolutive si MBT. Prof. Gheorghe este, de asemenea, conferentiar la Universitatea Pitesti.
- contactele internationale stabilite de catre aplicant la: SAP Research, TU München (Prof. Javier Esparza), TU Berlin (prof. Ina Schieferdecker), Univ. din Edinburgh (Prof. Colin Stirling), Universitatea din Boston (Prof. Calin Belta), Université de Franche-Comté (Prof. Bruno Legeard), Universitatea din Konstanz (Prof. Stefan Leue), si multe altele.

Riscuri posibile si solutii alternative

Ca in orice proiect indraznet, exista posibile riscuri:

- **Risc pentru Obiectivul (a):** Combinatia intre profilele UML pentru SOA si testare este artificiala si ineficienta.

Solutie: Aplicantul va discuta despre problemele intampinate cu experti de top in domeniu, precum Prof. Ina Schieferdecker, co-autor al unei carti Springer despre profile de testare UML si co-autor al standardului U2TP. In cazul in care persista inca dificultati, problemele vor fi documentate in articole de conferinta pentru informa comunitatea despre ele.

- **Risc pentru Obiectivul (b):** Abordarile evolutive dezvoltate nu sunt foarte eficiente pentru MBT si SOA.

Solutie: Din nou, experti in domeniu (prof. Ipate, prof. Gheorghe, prof. Legeard) vor fi abordati pentru a afla daca problemele sunt de natura fundamental sau datorate doar lipsei de experienta mare a aplicantului in ET. Mai mult decat atat, chiar daca aplicantul nu a publicat inca in domeniul ET, are experienta cu tehnici similare bazate pe cautare precum model-checking directionat (directed model-checking - colaborare cu Prof. Leue) si sinteza strategiilor de control a robotilor (colaborare cu Prof. Belta). In plus, exista solutii alternative din domeniul tehnicilor de cautare meta-euristica precum *particle swarm optimization* sau *simulated annealing*.

- **Risc pentru Obiectivul (c):** Rezultatele cercetarii nu scaleaza pe exemple practice.

Solutie: Aplicantul are multa experienta in cercetarea industriala. Validarea pe exemple reale se va face de la bun inceput in proiect (prin semnarea cu SAP de acorduri de nedezvaluire – *non-disclosure agreements*). In plus, aplicantul are experienta cu mai multe produse MBT cum ar fi TestDerivator de la IBM Research, SpecExplorer de la Microsoft sau TestDesigner de la Smartesting conceput de Prof. Legeard.

- **Risc pentru intreg proiectul:** Sunt prea multe obiective propuse spre abordare in interval de 2 ani.

Solutie: Desi par a fi multe lucruri de abordat, in primul rand exista o mare experienta a aplicantului cu privire la temele studiate si in al doilea rand, exista o stransa legatura intre diferitele obiective, ceea ce le face potrivite pentru a fi studiate impreuna.