17th Dutch Testing Day

Testing Evolvability

November 29, 2011

University of Twente, Enschede

Formal Methods & Tools

CTIT Centre for Telematics and Information Technology

UNIVERSITY OF TWENTE.

NWO

Programme Committee

Axel Belinfante (Universiteit Twente)
Henk van Dam (Collis)
Hans-Gerhard Gross (Technische Universiteit Delft)
Bart-Jan de Leuw (Logica)
Iris Pinkster (Professional Testing BV)
Maurice Siteur (Capgemini)
Mariëlle Stoelinga (Universiteit Twente)
Jan Tretmans (ESI en Radboud Universiteit Nijmegen)
Tim Willemse (Technische Universiteit Eindhoven)

Steering Committee

Henk van Dam (Collis)
Jan Tretmans (ESI en Radboud Universiteit Nijmegen)

Organisation

Mariëlle Stoelinga (Formal Methods & Tools, Universiteit Twente)
Axel Belinfante (Formal Methods & Tools, Universiteit Twente)
Mark Timmer (Formal Methods & Tools, Universiteit Twente)

Table of Contents

Preface 4
Your host: Formal Methods and Tools 5
Programme 6
Quantitative Testing Using UPPAAL 7
TOPAAS-model 8
Constructing Formal Models through Automata Learning 9
Creating automated tests that survive even continuous change 11
Company descriptions 12
Managing the co-evolution of software artifacts 18
Risk Based testing, a piece of cake or not? 21
Testing of Highly Distributed Service-Oriented System using Virtual Environments 23
Evolution of Test Automation 25

Preface

Dear Testprofessional,

Welcome to the 17th Dutch Testing Day! I am very pleased to see you all here at the, recently completely renovated, campus of the University of Twente.

For me personally, the synergy between academic and industrial testing activities is one of the most attractive aspects of the Dutch Testing Day: we have participants, speakers, and financial contributions from both worlds, fostering cross-fertilization and building bridges between them. It is widely known that industrial-academic partnership is a key driver to innovation and to remain at the competitive edge.

Therefore, I would like to thank everybody who made the 17th Dutch Testing Day a success: First of all, many thanks go to all participants: without participants, no Testing Day. Also, I would like to thank everybody who submitted an abstract, giving the Programme Committee a hard job in selecting the 7 best abstracts for presentation.

Moreover, we are very much indebted to our sponsors: CIMSOLUTIONS, SQUERIST, AESTIS KMG, the Institute for Software Quality, ASML, Valori, the Centre for Telematics and Information Technology, NWO, Improve Quality Services, Better Be, Professional Testing, Refis and Collis. Their financial contribution allows the participants to attend the Dutch Testing Day free of charge and shows the importance of the topic.

I am grateful to Rector Magnificus Ed Brinksma and invited speaker Kim Larsen for, respectively, opening the testing day, and giving the keynote speech.

Finally, I would very much like to thank the other members of the organizing team of the 17th Dutch Testing Day: Axel Belinfante, Joke Lammerink and Mark Timmer for their fantastic support.

I wish you a fruitful day!

Mariëlle Stoelinga

Formal Methods & Tools
University of Twente

Your host: Formal Methods and Tools

Today, the Formal Methods and Tools group at the University of Twente is happily hosting you at the 17th Dutch Testing Day!
FMT provides education in the Bachelor and Master study of Computer Science. Its research is performed in the research institute CTIT.

The mission of FMT is *to improve the dependability of critical ICT infrastructure in our society*. To this end, we facilitate the modelling and analysis of software-intensive embedded systems. Based on models of both functional and quantitative system behaviour, we are able to predict the quality of systems and design better systems, also reducing development costs by eliminating faults at early design stages.

FMT develops scientific, mathematical theories, methods, algorithms and high-performance tool support for the design and analysis of software-intensive embedded systems. Our mission builds on extensive experience in concurrency theory (process algebra, automata, graph grammars, separation logic).
We have specialized in three focus areas:

(1) Applications to realistic **object-oriented models and software;**
UML models and Java code are transformed and analysed.
(2) Treatment of **quantitative aspects** of embedded systems;
Real-time, probabilities, performance, availability, physical environment.
(3) **High-performance model checking tools** to verify at industrial scale;
We use symbolic methods, multi-core programming, and GRID computing.

Of course, also **automated model-based testing** plays an important role in our research. Actually, FMT was one of the first groups to recognize that one can be scientific about testing! Already in the early nineties, Ed Brinksma – currently our Vice-Chancellor – started to develop the testing theory for distributed protocols, solving the problem of inherent non-determinism. A well-known school of model-based testing was built around Jan Tretman's *ioco-theory*, leading to the widely used research tool (J)TorX by Axel Belinfante. This tool was used in testing a Storm Surge Barrier! Our student Machiel van der Bijl co-funded his own spin-off company Axini on testing. Nowadays, Mariëlle Stoelinga and Axel Belinfante provide research and education on model-based testing.

We apply our methods in projects with industrial and academic partners. An important safety-critical application domain consists of *Interlockings for railway signalling*. Other application domains include wireless sensor networks, embedded and distributed control systems, smart card security, and certified technology for software transformation.
Currently, we also investigate the application of our techniques to the concurrency found in living cells, as studied in molecular biology. Besides understanding life, the ultimate aim here is to have model-driven methods for rational drugs design.

Prof. Jaco van de Pol
Group Leader of FMT

Programme

09.30-10.10	**Registration and coffee**
10.15-10.30	**Opening** Ed Brinksma
10.30-11.30	**Keynote** Kim Larsen (University of Aalborg, and Center for Embedded Software Systems)
11.30-12.00	**TOPAAS-model** Ed Brandt (Refis)
12.00-12.30	**Constructing Formal Models through Automata Learning** Fides Aarts, Faranak Heidarian and Frits Vaandrager (Radboud University Nijmegen)
12.30-14.00	**Lunch**
14.00-14.30	**Creating automated tests that survive even continuous change** Martin Gijsen (De Analist)
14.30-15.00	**Managing the co-evolution of software artifacts** Martijn Klabbers, Alexander Serebrenik, Joost Gabriels (LaQuSo & TU Eindhoven)
15.00-15:30	**Risk Based testing, a piece of cake or not?** Jeanne Hofmans (Improve Quality Services)
15.30-16.00	**Coffee break**
16.00-16.30	**Testing of Highly Distributed Service-Oriented Systems using Virtual Environments** Faris Nizamic, Rix Groenboom, Alexander Lazovik (University of Groningen, Parasoft)
16.30-17.00	**Evolution of Test Automation** Jan de Coster (Micro Focus)
17.00	**Drinks**

Quantitative Testing Using UPPAAL

Kim G. Larsen

UPPAAL is a tool for modeling, simulating and verifying real-time and hybrid systems, developed collaboratively by Deparment of Computer Science at Aalborg University and Department of Computer Systems at Uppsala University since the beginning of 1995 (see www.uppaal.com).

Since 2003 significant effort has been put into adapting UPPAAL towards testing of quantitave aspects of embedded software. The talk will present the usage of the UPPAAL branches CORA and TIGA to off-line test-case generation with guaranteed coverated of a given design model, as well as the application of the branch TRON to efficient on-line generation of execution. The talk will report on industrial experiences and take-up as well as efforts towards fitting the UPPAAL technology with existing commercial tools, e.g. HP TestDirector, Rational Rose and Simulink.

Finally, the talk will present the new simulation based engine of UPPAAL, SMC, and indicate how this will allow for analysis and testing of refined performance properties of embedded systes, such as expected response- and execution-times, and how we will pursue this line within the newly started ARTEMIS project MBAT (Model-Based Analysis and Test).

About the presenter

Kim Larsen is a Professor in the Department of Computer Science at Aalborg University within the Distributed and Embedded Systems Unit and director of the ICT-competence center CISS, Center for Embedded Software Systems. He is also director of DaNES, Danish Network for Intelligent Embedded Systems, an Advanced Technology Platform, and the Innovation Network InfinIT. Finally he is co-director of the VKR Center of Excellence MT-LAB and will be the director of the new Danish-Chinese Basic Research Center IDEA4CPS.

His research interests include modeling, verification, performance analysis of real-time and embedded systems with applications to concurrency theory and model checking. In particular he is prime investigator of the real-time verification UPPAAL as well as the various new branches of the tool targeted towards optimization, testing, synthesis and compositional analysis.

TOPAAS-model

Ed Brandt (Refis)

The Dutch Directorate for Infrastructure and Water management is currently introducing probabilistic management to storm surge barriers and other objects. A risk analysis is the centre of this approach and gives direction to test intervals, maximal time to repair and modifications to the object. Ssoftware failure is also a component in this. Estimating software failure using existing methods however appeared to be not very reliable.

In order to address this problem, an expert group consisting from Det Norske Veritas, Rijkswaterstaat, Movares, Eindhoven University of Technology, Logica Mission Critical Systems, Refis Reliability Engineering and Intermedion have developed the TOPAAS[1] method. The method describes both the usage of failure tree analysis as the estimation of software failure and combines, besides the scientific knowledge, the experience of software reliability assessment in the field of nuclear safety, process industry, transport safety, avionics and primary water defences.

The principle idea of TOPAAS is that software can be divided into modules of which failure probability can be modelled in a failure tree. Failures can either be caused by unexpected input or by defects in logic and algorithms. Estimation of failure probability is based on a Bayesian approach using the expert judgement on a parametermodel with the following dimensions:
a) Developmentproces
b) Product
c) Requirements traceability
d) Testing
e) Operational use

The method has been calibrated using twenty reference projects. The overall conclusion is that the results of the model show a very strong correlation with the estimation by experts. And therefore that the method is a fairly reliable estimation of software failure probability.

About the author

Ed Brandt is working in information technology since 1982. He specialized in software testing since 1996 and founded Refis in 2003, specialized in reliability engineering and software metrics. He is one of the authors of the Topaas-method.

[1] Task Oriented Probability of Abnormalities Analysis for Software

Constructing Formal Models through Automata Learning

Fides Aarts, Faranak Heidarian, Frits Vaandrager

Model-based system development is becoming an increasingly important driving force in the software and hardware industry. The construction of models typically requires specialized expertise, is time consuming and involves significant manual effort, implying that in practice often models are not available, or become outdated as the system evolves. In practice, 80% of software development involves legacy code, for which only poor documentation is available. Manual construction of models of legacy components is typically very labor intensive and often not cost effective. The solution is to infer models automatically through observations and test, that is, through black box reverse engineering.

State-of-the-art tools for active learning of state machines are able to learn state ma- chines with at most in the order of 10.000 states. This is not enough for learning models of realistic software components which, due to the presence of program variables and data parameters in events, typically have much larger state spaces. Abstraction is the key when learning behavioral models of realistic systems. Hence, in most practical applications where automata learning is used to construct models of software components, researchers manually define abstractions which, depending on the history, map a large set of concrete events to a small set of abstract events that can be handled by automata learning tools. We show how such abstractions can be constructed fully automatically for a class of extended finite state machines in which one can test for equality of data parameters, but no operations on data are allowed. Our approach uses counterexample-guided abstraction refinement: whenever the current abstraction is too coarse and induces nondeterministic behavior, the abstraction is refined automatically. Using a prototype implementation of our algorithm, we have succeeded to learn fully automatically models of several realistic software components, including the biometric passport and the SIP protocol.

References:

1. F. Aarts, B. Jonsson, and J. Uijen. Generating models of infinite-state communication protocols using regular inference with abstraction. In A. Petrenko, J.C. Maldonado, and A. Simao, editors, 22nd IFIP International Conference on Testing Software and Systems, Natal, Brazil, November 8-10, Proceedings. IFIP, 2010.
2. F. Aarts, J. Schmaltz, and F.W. Vaandrager. Inference and abstraction of the biometric passport. In Proceedings 4th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2010), 18-20 October 2010
3. F. Aarts and F.W. Vaandrager. Learning i/o automata. In P. Gastin and F. Laroussinie, editors, 21st International Conference on Concurrency Theory (CONCUR), Paris, France, August 31st - September 3rd, 2010, Proceedings, volume 6269 of Lecture Notes in Com- puter Science, pages 7185. Springer, 2010.

About the authors

Fides Aarts is a PhD student at the Radboud University in Nijmegen. After she has finished her Master thesis in Uppsala on inference and abstraction of communication protocols under supervision of Bengt Jonsson, she continued working on learning automata. Currently, she is involved in automatically generating abstractions of automata to extent inference to systems with large parameter domains.

Faranak Heidarian is a last year PhD student in Radboud University Nijmegen. She works in MBSD group and her research is about abstraction refinement. She a has bachelor degree in software engineering, and a Masters degree in Computer Science from Sharif University of Technology, Tehran, Iran.

Frits Vaandrager has a strong interest in the development and application of theory, (formal) methods and tools for the specification and analysis of computer based systems. In particular, he is interested in real-time embedded systems, distributed algorithms and protocols.
Together with Lynch, Segala, and Kaynar he developed the (timed, probabilistic and hybrid) input/output automata formalisms, which are basic mathematical frameworks to support description and analysis of computing systems. He has been and is involved in a large number of projects in which formal verification and model checking technology is applied to tackle practical problems from industrial partners. Within the OCTOPUS project with Oc\'{e} he is currently involved in the construction of the DSEIR toolset for model-driven design-space exploration for embedded systems. Recently, he has also become very interested in automata learning.

Creating automated tests that survive even continuous change

Martin Gijsen

While it may be more obvious during development than in the maintenance phase, software evolves all the time. Automated system and acceptance tests often respond to these changes by refusing to run any longer. Because of the maintenance effort, many projects abandon automated testing after a while or do not even attempt it, and miss out on its many benefits. Which is even more unfortunate considering that the more changes affect the software, the more important the automated test is to make sure the software still works as before.

This presentation discusses practical techniques that have reduced maintenance effort significantly, using real examples from their application to two systems: A web application and an embedded system. These techniques are:
- Moving interface details out of test cases,
- Moving execution details out of test cases,
- Moving tooling details out of test cases,
- Moving environmental / configuration data out of test cases,
- Moving test data out of test cases,
- And (test) specification techniques from (Agile) approaches like BDD (Behavior Driven Development).

Applying these techniques has resulted in automated testing solutions that can deal with the evolution of the software. And while they are not new, they are rarely discussed, so many organisations are not familiar with them or are not yet applying them to their testware.

Lessons

The main lessons from this presentation are:
- That certain techniques reduce maintenance to the testware for a system, regardless of its kind of interface or what tooling is used,
- That these techniques boil down to moving maintenance sensitive details elsewhere using abstractions, and
- How to apply the techniques.

About the author

Martin Gijsen is an independent test automation architect. He designs and implements low maintenance solutions for automated testing that suit an organisation in terms of policies, people, processes and technology, for any kind of system. Martin is also a trainer and coach, a frequent presenter at conferences and the author of the PowerTools, an open source framework for automated testing.

Company descriptions

AestisKMG, het vroegere QED, stelt zich als doel het inzichtelijk maken van de kwaliteit en risico's bij het in gebruik nemen van informatiesystemen met een sterke focus op de business. Wij zijn actief bij zowel de startfase van ontwikkel- en testprocessen, als bij de acceptatie van op te leveren producten.

http://www.aestiskmg.nl

CIMSOLUTIONS is een ICT-Dienstverlener op het gebied van administratieve en industriële automatisering, opererend vanuit onze 6 vestigingen in Nederland. We zijn ISO-9001:2008 en NEN4400-01 gecertificeerd voor consultancy, detachering en projectuitvoering en 'Top Employer ICT'. Onze uitdaging is om onze klanten succesvol te laten zijn in hun (test)projecten.

http://www.cimsolutions.nl

IfSQ, Institute for Software Quality is a group of software professionals united by a common goal: to provide an anchor for code inspection in the software development process by producing, publishing and promoting a coherent, transparent set of Standards for Computer Program Source Code for software developers and their clients.

http://www.ifsq.org

Squerist is gespecialiseerd in de kwaliteit van bedri jfsprocessen en ICT. Onze testprofessionals zijn getraind in het specificeren, uitvoeren en begeleiden van alle testsoorten en testvormen, zowel handmatig als geautomatiseerd. U mag verwachten dat wij verder kijken dan alleen onze opdracht om te komen tot een goed eindresultaat. Wij leveren meer dan alleen mensen, wij leveren commitment.

http://squerist.nl

AESTIS KMG

Aestis[KMG] is actief binnen acceptatietesten en de kwaliteitszorg van ICT-systemen en producten, met een sterke focus op de business. Naast de gangbare acceptatietestvormen zijn wij van mening dat testen en kwaliteit verder gaat dan ICT. Succesvolle implementatie wordt immers bepaald door de combinatie van kennis, businessprocessen en ICT. Daarbij dient de vraag te worden beantwoord of, naast de ICT, ook de organisatie zelf klaar is voor het in gebruik nemen van de (nieuwe) systemen.

Succesvolle implementatie gaat niet alleen over ICT...

Net zo belangrijk is de vraag of de organisatie er klaar voor is.

Meer weten ?

AestisKMG
Wormerhoek 14k
2905 TX
Capelle aan den IJssel
Tel.: 010-2842424

info@aestiskmg.nl

CIMSOLUTIONS
Automation for Industry & Business

Learn, create and make it work!

Join *Our* Team !!!

Studeer je binnenkort af of ben je net afgestudeerd?
Salaris € 30.000 tot € 35.000 plus auto en onkostenvergoeding

CIMSOLUTIONS Weekend in Londen 2010

CIMSOLUTIONS daagt je uit om lid te worden van een uniek team van ICT professionals.

Wij zoeken: HBO en Universitair geschoolde mensen met een voorliefde voor ICT die zich willen ontwikkelen tot top professionals.

Wij bieden: een trainingsprogramma, gevarieerd werk en uitdagende projecten in een collegiale, informele en professionele werksfeer.

Interesse in een functie bij CIMSOLUTIONS? Stuur je sollicitatiebrief met CV naar onderstaand adres of mail naar **hrm@cimsolutions.nl.**

Voor meer informatie kun je onze website bezoeken via **www.cimsolutions.nl** of bellen met Jos Peek, tel.: 0347-368100 tot 21.00 uur.

Vianen | Best | Deventer | Rotterdam | Amsterdam | Groningen | Dhaka

CIMSOLUTIONS B.V. | Havenweg 24, 4131 NM Vianen | Postbus 183, 4130 ED Vianen | The Netherlands
Phone: (+31) 347-368100 | **Fax:** (+31) 347-373777 | **E-mail:** cimsolutions@cimsolutions.nl | **Internet:** www.cimsolutions.nl


GEVAAR
VERBODEN WERKING TE STELLEN
Is your software up to standard?
IFSQ Level-1
WIP-1
WIP-2
WIP-3
SP-1
SP-2
SPM-1
www.ifsq.org
ifsq
STANDARDS FOR COMPUTER PROGRAM SOURCE CODE
www.ifsq.org

Managing the co-evolution of software artifacts

Joost Gabriels, Danny Holten, Martijn Klabbers, Wiljan van Ravensteijn, Alexander Serebrenik

Software development projects are virtually always carried out under pressure. Planning and budgets are tight, room for errors is non-existent and the pressure to deliver is high. Natural questions for (test) managers arise, such as: “When have we tested enough?” and ”How many tests do we have to redo for this new version?''. The naive answer would be: “when we have convinced ourselves through testing that all requirements are satisfied.”. Unfortunately, attaining maximal confidence with minimal effort is not easy.

In order to convince ourselves that the system does what it is supposed to do, tests are required. Requirements, design and code change during the development of software. As a consequence, tests need to change as well for in the end we want to ensure that all requirements and risks are adequately addressed with tests. For this, tests at different levels of abstraction and for different software artifacts are required and need to be managed.

To relate user requirements, design, code and tests, traceability matrices are often used. Traceability allows to link elements from different software artifacts, like requirements, design components and code components, to each other and to test cases. As a result, traceability can be used to analyze for example how well software artifacts are covered by test cases. Because a requirement leads to design components and eventually to code, tests are needed at each stage. Traceability can tell us how well test cases cover different software artifact elements. This information can be used to uncover mistakes in software artifacts at an early stage and actively manage the development and test efforts. Unfortunately, traceability information is often spread out over multiple artifacts and describes only the current situation.

TraceVis, a visual analytics tool based on the master thesis of Van Ravensteijn[2], combines the traceability information of multiple software artifacts in an interactive way. Furthermore it provides a way of assessing the evolution of traceability between artifacts though a timeline. Figure 1 shows the traceability between four, vertically placed, hierarchical software artifacts: acceptance test plan, user requirements document, software requirements document, and architectural design document. Each line represents a link between elements of two artifacts. Priorities and/or risks can be marked with colors and hierarchies can be collapsed and extended. Furthermore, the edge bundling technique bundles similar relations in the middle, clearly showing deviations.

[2] W.J.P. van Ravensteijn, *Visual traceability across dynamic ordered hierarchies*, 2011, Eindhoven University of Technology

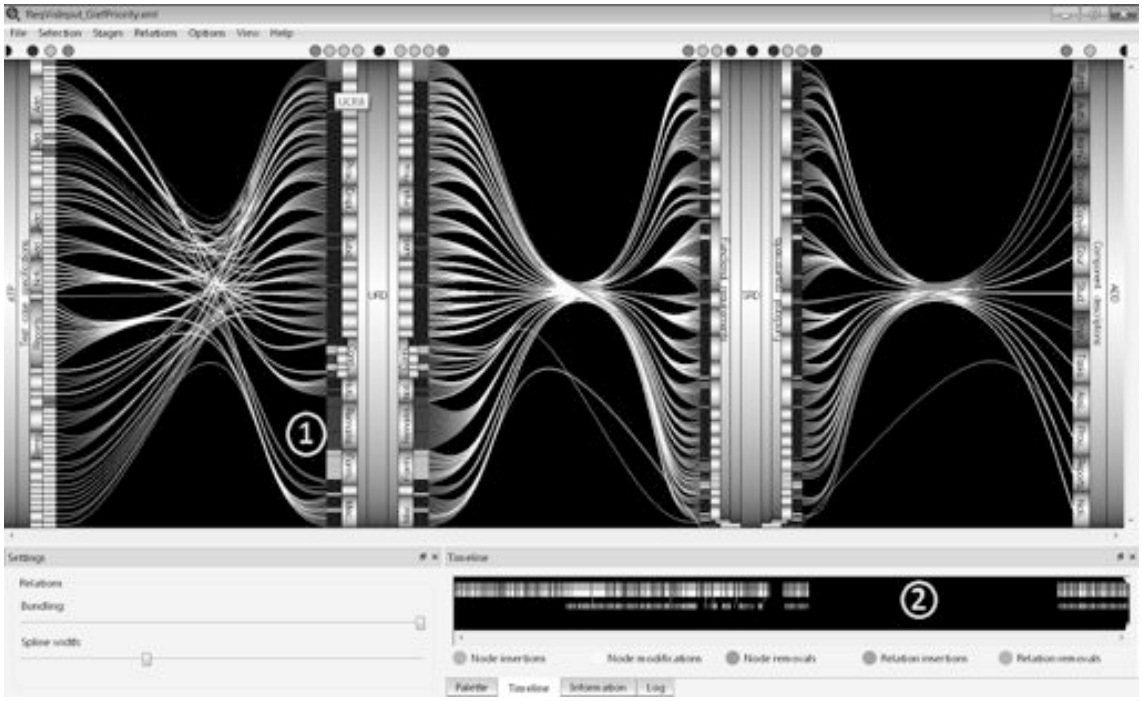


Figure 1. TraceVis tool with traceability information from a student capstone project at the Eindhoven university of Technology. The edge bundeling technique makes it easy to spot deviations. The gap labeled (1) shows some medium and low priority user requirements not covered by acceptance tests. The gap labeled (2), located in the timeline, shows that test cases were added very late in the project. The gap itself relates to implementation activities in which there were no changes to the shown software artifacts

Already at first glance, we can see points of attention in Figure 1: a gap in requirement coverage and a gap in the timeline.

The evolutionary traceability information allows us to see how well tests cover artifacts and whether risks are sufficiently tackled. It gives insight in the balance between tests, priorities, and risks and can support decision making in assigning test effort. Furthermore, it can help in determining which tests need to be redone when a certain component or requirement changes. The insight in the co-evolution of software artifacts and associated tests makes it possible to actively manage test effort from an early stage on.

About the authors

Joost Gabriels (j.m.a.m.gabriels@tue.nl, +31 (0)40 247 5886) received his M.Sc. in Computer Science from the Eindhoven University of Technology in 2007. Currently, he is a researcher and consultant at the Laboratory for Quality Software, Eindhoven University of Technology. In his work, he addresses mostly industrial problems in the area of software and system quality. His interests cover a broad area, including software architecture and design, specification languages, formal methods, program verification and the quality of software in general.

In 2005, *Danny Holten* (danny.holten@synerscope.com) received his Master's degree in computer science (with honors) from the Dept. of Mathematics & Computer Science at Eindhoven University of Technology (TU/e). From 2005 to 2009 he worked on an NWO-funded PhD project at the same university under the

supervision of prof.dr.ir. Jarke J. van Wijk. This research was part of the SWERL Reconstructor project, for which he developed techniques for the visualization of graphs and trees to aid in program understanding tasks; this led to his PhD dissertation in 2009. He received the best paper award at IEEE INFOVIS 2006 for the work on Hierarchical Edge Bundles and was nominated for the best paper award at ACMCHI 2009 for the joint work on the visualization of directed edges in graphs. He furthermore received the TU/e Doctoral Project Award (part of the TU/e Academic Awards 2010) for the best PhD dissertation at Eindhoven University of Technology in 2009. From 2009 to 2011, he worked as a postdoctoral researcher at the TU/e Dept. of Mathematics & Computer Science as well as within LaQuSo, the TU/e Laboratory for Quality Software, on the NWO-funded "Expression of Interest" (EoI) project under the supervision of prof.dr.ir. Jarke J. van Wijk. The aim was to study how interesting data aspects can be visualized in a clear way, which led to the development of generic models and guidelines. These formed the basis for a number of new visualization methods and techniques. Since April 2011, Danny Holten is CTO at SynerScope B.V., a Dutch visualization-research-inspired TU/e spin-off company that leverages his PhD research for "Big Data" analysis.

Martijn Klabbers (m.d.klabbers@tue.nl, +31 (0)40 247 3519) received a M.Sc. in Computer Science from the Technical University Delft, The Netherlands in 1995. At the Technical University of Eindhoven, he is a senior consultant at LaQuSo (Laboratory for Software Quality). In the past, Martijn researched decision support models at the TUE. Before LaQuSo, he was successively software developer, project manager, and product manager at NIPO Software. His research interests include certification, decision support systems, and user requirements.

In 2011, *Wiljan van Ravensteijn* (wiljan.van.ravensteijn@synerscope.com) received his Master's degree in computer science (with honors) from the Dept. of Mathematics & Computer Science at Eindhoven University of Technology (TU/e). His Master's thesis was titled "Visual traceability across Dynamic Ordered Hierarchies". Since August 2011, Wiljan van Ravensteijn is Software Developer at SynerScope B.V., a Dutch visualization-research-inspired TU/e spin-off company that leverages Danny Holten's PhD research for "Big Data" analysis.

Dr. *Alexander Serebrenik* (a.serebrenik@tue.nl, +31 (0)40 247 3595) is an assistant professor of Model-Driven Software Engineering at the Eindhoven University of Technology (TU/e). He has obtained his Ph.D. in Computer Science from Katholieke Universiteit Leuven, Belgium (2003) and M.Sc. in Computer Science from the Hebrew University, Jerusalem, Israel. Dr. Serebrenik has co-authored more than one hundred scientific papers, has acted as a committee member and external reviewer of numerous scientific journals and meetings, as well as evaluator of a number of European, international and national project proposals. He currently co-supervises five Ph.D. students. Dr. Serebrenik's areas of expertise include software evolution, maintainability and reverse engineering, program analysis and transformation, process modeling and verification. Furthermore, Dr. Serebrenik has been involved in a series of industrial projects pertaining to software quality assessment.

Risk Based testing, a piece of cake or not?

Jeanne Hofmans

Motivation and novelty
Risk based testing is not a novelty at all. Based on the fact that it is impossible to test everything, the idea of focusing test effort on the most risky areas has been embraced by both testers and management. But even today many companies and projects are struggling with their risk based test approach. Especially since systems under test are becoming more complex the need to focus on the risky areas increases. And thus the need for a flexible and practical approach for identifying and managing risks.

Problem addressed and proposed solution
Many projects start off with a product risk analysis and promising words in the testplan. During most projects the risk analysis disappears in the background. The set of executed test cases is not evidently based on risk. Testers, management and stakeholders cannot see whether the test approach was appropriate considering the risks involved; risk based reporting becomes impossible. Projects that must deal with continuous change do not benefit by a complex product risks analysis.

A practical solution is to base the product risk analysis on three types of product risk:
• Specification;
• Integral;
• Regression.

These three types of product risk are very much in line with the way testers (should) test. First of all the specification is tested, where more attention is paid to the parts which are identified as most risky. Then special attention is paid to risks that exist, but are not clear in the specification; the integral risks. Finally attention must be paid to the approach for regression testing.

During the (test)project the risks, either being specification, integral or regression, are used as a basis for test plans, test design & execution and reporting. Of course new risks can be added at any time. The strength of this approach is that is uses a combination of risk analysis techniques: FMECA, brainstorm, PRISMA. Besides it can be used in addition to other risk analysis techniques like PRIMA.

Lessons learned
The identification of three types of product risk is basically the result of lessons learned in practice. Because the three types of risk are in line with the way testers work, it's very practical for testers to use. It's also very appealing to managers and stakeholders because the test process finally becomes transparent to them.

About the author

Jeanne Hofmans has studied Software Technology at Utrecht University and now is a test consultant at Improve Quality Services. She has participated in many projects in both the financial and the embedded domain where risk based testing was more or less successful. The suggested solution was implemented by Jeanne at a technical test center and presented at Testnet Spring Event (Nieuwe helden) and at a Supervision session at Rabobank International. Momentarily Jeanne works as a test manager for the Sluiskiltunnel and as an auditor for several other tunnelsprojects.

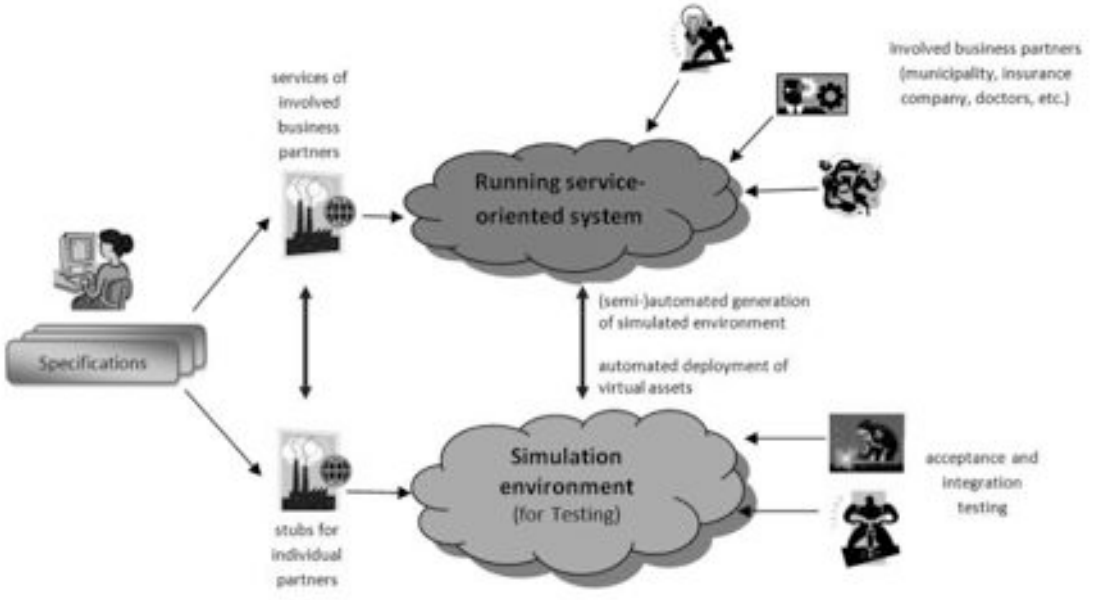
Testing of Highly Distributed Service-Oriented System using Virtual Environments

Faris Nizamic, Rix Groenboom, Alexander Lazovik

Modern application landscapes are becoming more and more interconnected. The introduction of SOA and ESB technology has increased the dependency on other systems, both internal and external. This has severe consequences for testing SOA applications, their availability, and the constraints on different environments.

Performing testing while the system is running may potentially cause a pollution of the system with test-data, sending out the test data to the end-users or even result in additional costs by invoking services that cost money per invocation. Testing costs get even higher if we consider that deploying a new version of software to different environments requires high expertise of system engineers that usually handle this kind of work. Often, the problem is that some external service is simply not available or it is available for just limited time for testing. All of these issues could be avoided if a simulation environment, behaving in the very same way as the real system, is used. Nowadays, generation of this kind of simulation environments is very feasible and it is something that needs to be explored further.

In this work we concentrate on how testing of web services is performed in highly distributed service-oriented systems using a simulation environment so that we can avoid some of the mentioned issues, or reduce their effects. To tackle this kind of problems, we propose to completely virtualize the real service-oriented system behavior by building the simulation environments (semi-)automatically. The goal is to perform the integration and acceptance testing without influencing the real system or making additional costs. In our research we especially emphasize how to make, manage and control simulation environment. In that sense, additionally to an automatic generation of virtual assets, we focus on automation of deployment process of the virtual assets and simplifying the switching between different environments (Production, Testing, etc.). On this level, we want to ensure that test-data is consistent across all virtual assets.



In order to demonstrate how this could work in practice, we present a simulation environment of real business process for the implementation of Wet Maatschappelijke Ondersteuning Law (WMO). The WMO is a Dutch law for supporting people that have a chronic disease or disability, so that these people can independently live in their homes and actively take part in everyday life despite their physical limitations. The support that is provided by the WMO typically includes transportation, a wheelchair, or a home modification (e.g. removing door posts for people using wheelchairs). The responsibility for a WMO request lies with the Dutch municipalities; they handle the complete business process. The process accesses external parties, e.g., insurance companies and doctors for medical advice.

In our previous work, we have demonstrated how simulation of service-oriented architectures using WS-BPEL can contribute to the decrease of integration testing costs on running service-oriented architectures. Now, also as a continuation of earlier work presented at the Dutch testing day in 2008 ("Tool supported collection of test data from multiple backend systems in a SOA architecture"), we plan to go even further and ease the testing by complete virtualization of the application behavior and deployment process automation.

About the authors

Faris Nizamic (f.nizamic@rug.nl) is PhD student at the University of Groningen (RuG). His research interests cover the areas of testing of highly distributed service-oriented systems. He holds a MSc in Computing Science from the University of Sarajevo, Bosnia and Herzegovina. Last two years he has worked as a Senior Quality Assurance Engineer in NAVTEQ with the main focus on testing of web services and automation of testing processes.

Rix Groenboom (rix.groenboom@parasoft.nl) is consultant for Parasoft in the area of testing and virtualization of modern SOA, SaaS and cloud-based architectures. He has written a large number of technical articles and presented on many IT industry conferences in Europe and the USA. His core area of expertise is specification, design and validation of software applications. He holds a MSc and PhD in Computing Science from the University of Groningen and published a thesis focusing on the formalization of domain knowledge.

Dr. Alexander Lazovik (a.lazovik@rug.nl, http://www.cs.rug.nl/~lazovik) is an assistant professor at Distributed Systems group at University of Groningen (NL), after being ERCIM fellow at CWI (NL) and INRIA (F), intern at IBM TJ Watson. He obtained the PhD in computer science from the University of Trento (I) in 2006. His research interests are in the areas of service-oriented and distributed computing, including service composition, business process validation and execution, diagnosis and automated repair using techniques such as automated planning and constraint-solving. Since last few years, he is also actively interested in pervasive and ubiquitous computing, with an emphasis on dynamic coordination of context-aware embedded devices.

The research is supported by the NWO SaS-LeG project, http://www.sas-leg.net, contract No. 638.001.207.

Evolution of Test Automation

Jan de Coster

Professional testers will know what you mean if you mention the Keyword driven testing method. It has become a common property and is used by organizations to prevent the disadvantages of Record/playback test automation. As applications may need thousands of automation keywords, Keyword driven testing also brings disadvantages in maintenance and complexity to the table. This means that there are still development opportunities. Where are we now in the evolution of test automation?

Evolution does not stop at Keyword driven testing. State driven testing is the next step in the evolution of test automation. Jan de Coster, testing expert at Micro Focus, has been involved in research around this new automated testing technique and the benefits that it brings.. Besides internal research, State driven testing is now used by several organizations to help them bring more structure in their automation efforts.

What characterizes this method? State driven testing categorizes keywords based on the situation. In many situations the next step in a test scenario is limited to a number of possibilities and the model anticipates on that. It knows what keywords follow logically. The use of this method results in, significant time savings in the construction and maintenance of the test scenario's. In this session, Jan de Coster shows how State driven testing avoids the difficulties of Keyword driven testing through an application state-transition model.

About the author

Jan de Coster is a testing guru with over 16 years of IT experience. He currently holds the position of Subject Matter Expert in Quality Solutions at Micro Focus. In this role, Jan provides domain expertise and thought leadership around Software Quality Assurance as well as delivering consultancy and project services into the field throughout Europe.

Jan started his career in software development and project management. However, it soon became clear that he had a passion for the qualitative aspect of software delivery and for the last 12 years he has specialized in integrating processes and tooling to help organizations increase quality in their software. More recently, Jan has been more closely exploring quality in relation to Agile Test Management and Agile Test Automation – areas, he feels, of significant importance to the future success of software delivery.

Sponsored by

Nederlandse Organisatie voor Wetenschappelijk Onderzoek