

Geração eficiente de casos de teste baseado em modelos IOLTS

Wallacy Sebastian Aparecido Jeronimo de Almeida¹, Adilson Luiz Bonifácio¹

¹Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina – PR – Brasil

wallacy.sebastian@uel.br, bonifacio@uel.br

Abstract. *While systems modernize, demand for systems capable of keep processes constantly executing, without errors, increase, especially critical systems. In this way, to reduce or remove errors in software development, it is necessary to consider testing. There are many algorithms to generate tests cases to detect erros in general, using formalisms as Input Output Labeled Transition System (IOLTS) uses. Other ways to test IOLTS reactive models is through algorithms that check conformance between the possible system implementations and its related specifications. As testing generation can result into infinite set of tests, in general, this process becomes impractical, requiring some alternatives. Therefore, this work intends to offer new algorithms to improve efficiency and effectiveness of testing generation in existing practical tools.*

Resumo. *Conforme os sistemas se modernizam, a demanda por sistemas capazes de manter processos em constante execução, sem falhas, aumenta, principalmente quando se trata de sistemas críticos. Dessa forma, a inserção de testes nos processos de desenvolvimento de um sistema se torna uma ação necessária para a diminuição dos riscos de falhas. Existem vários métodos para a geração de casos de teste para detecção de falhas, usando formalismos, tal como os Sistema de Transição Rotulada de Entrada e Saída (IOLTS). Outra forma de se testar modelos reativos IOLTS é através de algoritmos que verificam a conformidade entre as possíveis implementações de um sistema e suas respectivas especificações. Como a geração de casos de teste pode resultar em conjuntos de testes infinitos, em geral, este processo se torna infactível, exigindo algumas alternativas. Neste sentido, este trabalho pretende oferecer novos algoritmos para melhorar a eficiência e eficácia de geração de testes em ferramentas práticas existentes.*

1. Introdução

Os sistemas reativos, que são definidos pelo seu comportamento de reação à estímulos externos (como protocolos de comunicação, periféricos de um computador, beacons, etc.), possuem uma grande importância econômica atualmente, além de um vasto uso científico, e tem sido uma tarefa constante desenvolvê-los para a obtenção de mais qualidade de recursos [1].

No desenvolvimento de um sistema, uma etapa importante a ser considerada é a de testes. Alguns testes serão básicos, outros mais rigorosos, de acordo com seu nível de aceitação de falhas. Para sistemas críticos, em que falhas são inaceitáveis, a fase de testes é importantíssima [7].

Quando se trata de testes rigorosos, podemos citar os testes baseados em modelos (ou *Model-based Testing* - MBT), que busca verificar a conformidade entre a especificação do sistema e a implementação sob teste (*Implementation Under Test* - IUT) a partir de um modelo formal [1]. O teste baseado em modelo se encaixa melhor em um ambiente empresarial, pois além de tornar viável a automação dos testes, este se utiliza de modelos formais, os quais envolvem a modelagem de especificações, os comportamentos desejados e possíveis implementações futuras [8].

Podemos ainda direcionar os testes em relação ao que buscamos como resultado, definindo assim que tipo de modelo será utilizado. Por exemplo, se existe a necessidade de comparar o comportamento desejado em relação ao IUT, usa-se o teste de conformidade. Há também o teste de estresse, no qual descreve o desempenho do sistema em cenários extremos e o teste de performance, que avalia a velocidade que o sistema executa as tarefas dadas [8].

Dado que o teste baseado em modelo se utiliza de modelos formais compatíveis entre as especificações do sistema e a sua implementação, é necessário definir que tipo de formalismo se deseja adotar. Dentre os modelos formais existentes, os Sistemas de Transição Rotulada de Entrada e Saída (IOLTS) são os mais adequados para sistemas reativos assíncronos, devido à sua representação adequada deste tipo de sistema.

Após essa definição, é importante analisar qual o tipo de foco da testagem e quais critérios serão utilizados para a avaliação dos modelos, para definir algoritmos com heurísticas que melhorem a eficiência na etapa de geração e seleção dos testes [6].

Ainda nesta etapa, deve-se levar em conta a detecção de falhas que o modelo será capaz de realizar, e com qual precisão. Essas falhas podem ser classificadas em [4]: erros de operação, onde é possível tornar os modelos equivalentes apenas alterando a função de saída; erros de transferência, onde os modelos serão equivalentes se houver mudanças ou adições de transições entre estados do modelo; e excesso ou falta de estados, no qual existem estados em excesso ou faltantes entre os modelos, implicando assim na desigualdade entre a implementação e a especificação.

Os meios para detectar as falhas levam em conta o foco da testagem, visto que as combinações de entrada ou de estados são potencialmente infinitas. Dessa forma é interessante limitar o conjunto de estados e transições, e portanto, limitar o tipo de falha a ser detectado. Como exemplo, pode-se mencionar a cobertura de transição, que consegue detectar todos os erros de operação, mas não os de transição e os de excesso ou falta de estados [4].

O foco deste trabalho está em realizar um estudo sobre métodos que possibilitem o aprimoramento do desempenho na geração de testes, otimizando e viabilizando cada vez mais a utilização de modelos formais em testes.

2. Fundamentação Teórico-Methodológica e Estado da Arte

As demandas por software aumentaram consideravelmente, em especial devido a situação global pandêmica recente. Manter os processos funcionais, como segurança, estabilidade e integridade, tem sido um desafio contínuo. Uma das importantes atividades do processo de desenvolvimento de sistemas que pode garantir uma melhor qualidade dos produtos de software [7] são os testes.

Sabe-se que o setor de tecnologia de uma empresa, ou até mesmo empresas de tecnologia, necessitam cada vez mais de entregar produtos e serviços de qualidade de forma rápida e eficiente, o que exige também um rápido entendimento do sistema pelo profissional que irá testá-lo. Além disso, várias empresas exigem também vários tipos de testes, tais como teste de estresse, segurança e disponibilidade. A principal classificação está no tipo de teste que se quer realizar [10], de caixa preta, onde o testador não tem acesso a estrutura interna do sistema em teste, e/ou de caixa branca, onde a estrutura interna, tais como código e processos, é conhecida. Qualquer teste realizado entre o tipo caixa preta e branca é denominado de teste de caixa cinza, onde o profissional deve conhecer, por exemplo, a estrutura do sistema, mas não o código.

Cada tipo de teste apresenta uma série de problemas em sua aplicação. Nos testes de caixa branca as seguintes dificuldades são enfrentadas:

- necessidade do sistema estar implementado;
- remodelação dos testes, que são executados paralelamente, quando o sistema sofre uma atualização; e
- escassez de profissionais capacitados para uma série de atividades específicas, dado que este deve conhecer o processo e estrutura do sistema, como também as ferramentas de teste e utilizá-las com eficiência.

Por isso, os testes de caixa branca se tornam inconvenientes em muitas situações [8].

Em contra-partida, os testes de caixa preta, em conjunto com modelos formais, oferecem algumas vantagens fundamentais para a geração de testes e melhor aproveitamento de processos em uma empresa, pois além dos problemas descritos acima que podem ser evitados utilizando caixa preta, os modelos formais fornecem uma base sólida para geração algorítmica e automática de testes [10].

Ainda que cada vez mais existam empresas evidenciando a importância dos testes em seus processos, o desenvolvimento e a propagação desta área se torna indispensável, visto que os modelos formais são pouco difundidos, e há muitas empresas utilizando métodos inconsistentes para testes, levando a erros completamente evitáveis caso fossem utilizados métodos mais seguros e eficientes [8, 10].

Das ferramentas disponíveis para teste, temos a Everest, que torna o teste de conformidade possível através de modelos IUT (Implementation Under Test), sendo totalmente viável a reprodução de testes de caixa preta [5]. Constantemente, a ferramenta recebe atualizações, inclusão de módulos e aprimoramentos para a geração de modelos de teste [8, 2]. No entanto, os algoritmos que implementam o modelo IOLTS carecem de desempenho, dado que para a inclusão de parâmetros diferentes ou mais complexos, modelos maiores são gerados, exigindo mais processamento e tempo de execução.

Um meio para melhorar o desempenho dos algoritmos seria manter um equilíbrio entre o tamanho de um modelo, sua precisão e legibilidade, aumentando o nível de abstração enquanto preserva a possibilidade de uso dos parâmetros [9].

Outro meio seria limitar a redundância dos modelos gerados, já que para um mesmo número de parâmetros, com os mesmos valores, existem inúmeras possibilidades de modelos de teste a serem gerados, sendo muitos destes equivalentes entre si, tornando inviável executar todos eles.

3. Objetivos

Este projeto terá como objetivo principal melhorar a eficiência da geração de modelos de teste utilizando a ferramenta Everest, através do modelo formal IOLTS. Para isso, será necessário entender sobre os modelos reativos e suas particularidades, compreender o modelo IOLTS, e então desenvolver um algoritmo que elimine redundâncias nos modelos gerados.

Os objetivos específicos do projeto são:

1. Estudar as propriedades e restrições sobre os modelos reativos e seu impacto tanto na geração desses modelos quanto na avaliação dos métodos de teste.
2. Propor estratégias para a geração e seleção eficiente de modelos de teste que apresentem as propriedades relevantes;
3. Implementar e testar as estratégias de geração, conforme os critérios e as propriedades propostas;
4. Realizar comparações de desempenho entre os métodos de geração de testes;
5. Divulgar resultados obtidos com os experimentos realizados através da ferramenta implementada, bem como sobre as estratégias e algoritmos propostos.

4. Procedimentos Metodológicos

Com o intuito de buscar conhecimentos fundamentais sobre modelos reativos, será indispensável entender sobre máquina de estados finita (FSM), teste baseado em modelo e teste de conformidade. Para isso, o livro *Model Based Testing of Reactive Systems* [3] irá propiciar a base necessária de conhecimento.

O artigo *Testing Techniques* [10] também será de grande apoio para entender sobre modelos reativos e o modelo IOLTS.

Como o modelo IOLTS possui uma estrutura semelhante a uma FSM, a exploração de métodos de geração e seleção de testes como Transition Tour, Método W, Método W parcial, Sequências Distintas e Unique Input Output [6], originalmente desenvolvidas para sistemas cuja especificação está na forma de uma FSM, será de grande relevância para a obtenção de métodos que eliminem redundância ou melhorem a eficiência na geração e seleção de testes. Dependendo do teste desejado, um ou mais métodos podem ser utilizados, de forma combinada, com o propósito de se obter algo mais eficiente.

5. Cronograma de Execução

As atividades deste projeto estão previstas para serem realizadas durante o período planejado no cronograma na Tabela 1 e descritas abaixo:

1. Revisão bibliográfica dos conceitos relacionados aos testes baseados em modelos;
2. Estudo dos métodos de teste usando modelos reativos assíncronos;
3. Estudo e investigação sobre as propriedades e restrições de modelos IOLTS;
4. Definição das classes de modelos que representam IUTs;
5. Desenvolvimento de algoritmos de geração de modelos para avaliação de métodos e ferramentas de teste baseados em modelos assíncronos;
6. Implementação dos algoritmos propostos e desenvolvimento de uma ferramenta de apoio;

7. Realização de experimentos práticos;
8. Divulgação de resultados através de publicações.

Atividade	2022				2023				
	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai
1	•	•							
2	•	•	•						
3		•	•	•					
4			•	•	•				
5			•	•	•	•	•		
6					•	•	•	•	
7					•	•	•	•	
8						•	•	•	•

Tabela 1. Cronograma de atividades

6. Resultados Esperados

Espera-se que o desenvolvimento deste projeto contribua para o aprofundamento do conhecimento sobre os testes baseados em modelos reativos e suas propriedades, a disponibilização de dados referentes a eficiência e performance dos métodos de geração e seleção de modelos de testes e a viabilização de acesso a ferramentas que se utilizarão destes métodos.

As conclusões obtidas da etapa preliminar de estudos devem ser aplicadas ao desenvolvimento de um algoritmo com melhor desempenho para a geração de modelos de teste e seleção dos mesmos.

A etapa final do trabalho contempla o desenvolvimento de uma ferramenta de geração e seleção de testes que possibilitará a avaliação de algoritmos e métodos de teste.

7. Espaço para assinaturas

Londrina, 12 de setembro de 2022.

Wallacy S. A. Jerônimo de Almeida

Aluno

Adilson Luiz Bonifácio

Orientador

Referências

- [1] Adilson Bonifacio. Everest: An automatic model-based testing tool for asynchronous reactive systems. *Journal of Software Engineering Research and Development*, 9:1–14, 08 2021.
- [2] Adilson Bonifacio and Camila Gomes. A model-based testing tool for asynchronous reactive systems. 10 2020.

- [3] Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, and Alexander Pretschner. *Model Based Testing of Reactive Systems - Advanced Lecture*. Springer, 2005.
- [4] T.S. Chow. Testing software design modeled by finite-state machines. *IEEE Transactions on Software Engineering*, SE-4(3):178–187, 1978.
- [5] Everest. Everest: conformance verification on testing reactive systems. <https://adilsonbonifacio.github.io/everest-site/index.html>. Acessado em 01/09/2022.
- [6] Susumu Fujiwara, Gregor Bochmann, Ferhat Khendek, Mokhtar Amalou, and Abderrazak Ghedamsi. Test selection based on finite state models. *IEEE Trans. Software Eng.*, 17:591–603, 06 1991.
- [7] Anand Moorthy and Nilesh Vaidya. *Sector Analysis - Financial services*, pages 49–51. Capgemini, 2022.
- [8] C. P. G. D. NASCIMENTO. Módulo de geração de modelos iolts para suportar os testes em modelos reativos. 2021.
- [9] Sébastien Salva. Reverse engineering models of concurrent communicating systems from event logs. 10 2021.
- [10] Jan Tretmans. Testing techniques, formal methods and tools group. 2002.