

Seleção Automática de Casos de Teste de Regressão Baseada em Similaridade e Valores

Francisco Gomes de Oliveira Neto
Patrícia D. L. Machado ¹

Resumo: Técnicas de re-teste seletivo baseadas em modificações podem selecionar casos de teste redundantes e desconsiderar casos de teste importantes que exercitam fluxos críticos não cobertos por estas modificações. Para tratar este problema, este trabalho apresenta uma nova técnica de re-teste seletivo baseada em modificações que considera o valor de cada caso de teste e a similaridade entre casos de teste, a fim de selecionar os casos de teste mais importantes, reduzindo significativamente o tamanho do conjunto, sem afetar a capacidade de detecção de defeitos. Os estudos de caso realizados mostram que a técnica é capaz de reduzir o tamanho de 70% a 80%.

Abstract: Selective retesting techniques based on modifications may select redundant test cases as well as exclude important test cases that exercise critical flows that are not covered by the modifications. To handle this problem, this work presents a new technique based on modifications that consider the value of each test case and the similarity among the test cases in order to select the most important ones, reducing significantly the size of the suite while not compromising defect detection capability. Case studies show that the technique is able to reduce the size from 70% a 80%.

1 Introdução

Teste de regressão é uma atividade realizada em elementos modificados de um sistema, com o objetivo de capturar defeitos de regressão que, por sua vez, são defeitos inseridos devido à modificação do software desenvolvido [1]. No teste de regressão, são consideradas duas versões do sistema: a versão do sistema anterior às modificações (versão base), e a versão após as modificações (versão delta).

Um dos principais problemas do teste de regressão é o alto custo, pois, a cada modificação, casos de teste podem ser criados, enquanto outros passam a ser inválidos, dificultando cada vez mais a realização do teste. Para tratar este problema, pode-se aplicar técnicas de priorização de casos de teste em que uma ordem de execução (prioridade) é associada aos casos de teste a partir critérios como cobertura de linhas de código, fator de exposição de riscos, dentre outros, com o objetivo de revelar defeitos no início da execução. Outro exemplo de abordagem é a seleção de casos de teste de regressão (também denominada de *re-teste*

¹SPLab-UFCG, Av. Aprígio Veloso, 882. Campina Grande, Brasil
{netojin@copin.ufcg.edu.br} {patricia@computacao.ufcg.edu.br}

seletivo) [2] aplicada com o objetivo de reduzir o tamanho do Conjunto de Casos de Teste de Regressão (CCTR). A seleção difere da priorização por tornar possível a redução do conjunto. Com menos casos de teste para executar, o custo da execução pode ser menor.

O re-teste pode ser realizado em dois contextos: código e especificação. Dentre estes dois contextos, o código é o mais abordado [3], devido à facilidade de execução dos casos de teste obtidos. No entanto, a abordagem baseada em especificação gera a expectativa de se obter uma rastreabilidade mais precisa de casos de teste relativos à modificação de uma funcionalidade, visto que os casos de teste podem ser re-gerados a partir de um modelo. Dessa forma, diversas técnicas de re-teste seletivo foram propostas na literatura.

No entanto, grande parte destas técnicas realizam a seleção considerando apenas a cobertura de modificações [4], enquanto que o CCTR pode apresentar casos de teste não modificados que exercitem fluxos críticos que precisam ser testados [5]. Se considerarmos por exemplo, uma aplicação para controle de tráfego aéreo; é possível que modificações nas unidades de medidas (e.g. milhas ou quilômetros) do módulo de entrada afetem o cálculo de velocidade das aeronaves. Portanto, além de testar as novas entradas é importante retestar também todos os cálculos que já eram realizados pelo sistema.

Sob esta perspectiva, propomos a técnica *Weighted Similarity Approach for Regression Testing* (WSA-RT) que seleciona os casos de teste mais críticos dentre os que interagem com as modificações realizadas no sistema. Para isto, utilizamos uma abordagem baseada em valores, em que o testador atribui pesos² para determinar a importância dos casos de teste. Para lidar com o nível de sistema, utilizamos abordagens de Teste Baseado em Modelos (TBM) nos modelos de especificação para automatizar o processo de geração e seleção.

WSA-RT é inspirada na técnica *Weighted Similarity Approach* (WSA) proposta por Bertolino et al. [6] que utiliza funções de similaridade e pesos para a seleção de casos de teste. Enquanto WSA realiza a seleção entre os casos de teste de um mesmo conjunto, WSA-RT observa a similaridade entre dois CCTR de versões diferentes do sistema procurando identificar automaticamente os casos de teste modificados e importantes.

Este artigo estende a versão preliminar apresentada em [7] com uma descrição mais detalhada da técnica e os resultados de 2 estudos de caso adicionais. O restante do artigo está organizado como segue. A Seção 2 apresenta os fundamentos teóricos. WSA-RT é descrita na Seção 3 sendo ilustrada por um exemplo simples. A avaliação da técnica é apresentada por meio de estudos de caso na Seção 4, seguida pela seção de trabalhos relacionados. Por fim, as conclusões deste trabalho são apresentadas na Seção 6.

²Este peso, ou valor, se refere ao critério definido pelo testador para caracterizar um caso de uso, ou caso de teste, como importante, ou crítico. Por exemplo, um custo de correção de defeitos, risco de expor defeitos, dentre outros.

2 Fundamentação Teórica

Re-teste seletivo (do termo *selective retesting*, ou ainda *selective regression testing*), tem como principal objetivo a redução dos custos de testar um programa modificado, por meio do reuso de testes existentes e da identificação de trechos modificados do programa ou de sua especificação que devem ser testados [2]. Considerando que o sistema pode ser entendido como uma abstração de estado e transições, dentre as diversas modificações em um software, podemos encontrar a remoção de transições da aplicação que não são mais desejadas pelo usuário, mudanças nos rótulos do modelo e a adição de transições. Um dos principais desafios do re-teste seletivo é a seleção de um subconjunto representativo para testar a versão modificada [2]. Esta representatividade é verificada por meio das características do subconjunto selecionado (e.g. cobertura de requisitos, defeitos, dentre outros).

Diferente da seleção baseada em cobertura, a seleção baseada em similaridades procura maximizar a diversidade entre os casos de teste selecionados. Esta diversidade, por sua vez, é medida por meio de uma função de similaridade capaz de caracterizar a semelhança entre um par de casos de teste através de elementos dos casos de teste (transições, estados, etc.). Sob o contexto de TBM, as entradas para essas funções são casos de teste abstratos definidos a partir do modelo de especificação da aplicação testada. Diversas funções de similaridades já são usadas por técnicas presentes na literatura [6, 8, 9]. Em alguns destes estudos, é possível observar que a seleção de casos de teste menos similares pode aumentar a cobertura de defeitos detectados, pois diferentes trechos da aplicação são testados.

Por fim, técnicas de seleção podem ser baseadas em uma configuração neutra ou classificada por valor. Na primeira, todas as funcionalidades são consideradas como iguais. Na segunda, baseada na abordagem de valor proposta por Boehm [10], é necessário estruturar quais são os “valores” desejáveis para o software, como, por exemplo, os recursos utilizados para diferenciar a importância de casos de uso, ou requisitos, no produto desenvolvido. Ou seja, podemos quantificar quais entidades (funcionalidades, usos, etc.) são importantes.

3 *Weighted Similarity Approach for Regression Testing*

WSA-RT é uma técnica capaz de identificar os casos de teste que interagem com as modificações do modelo e então selecionar aqueles que são mais importantes (i.e. com o maior valor, ou peso). A Figura 1 apresenta as etapas para a execução de WSA-RT.

As modificações são identificadas a partir de uma *matriz de similaridades* que representa as similaridades entre duas versões distintas do software. O valor de similaridade é obtido comparando casos de teste de versões diferentes do sistema, que, por sua vez podem ter sido gerados automaticamente a partir dos modelos de especificação. São comparados os casos de testes da versão base e da versão delta, denominados, respectivamente, T e T' .

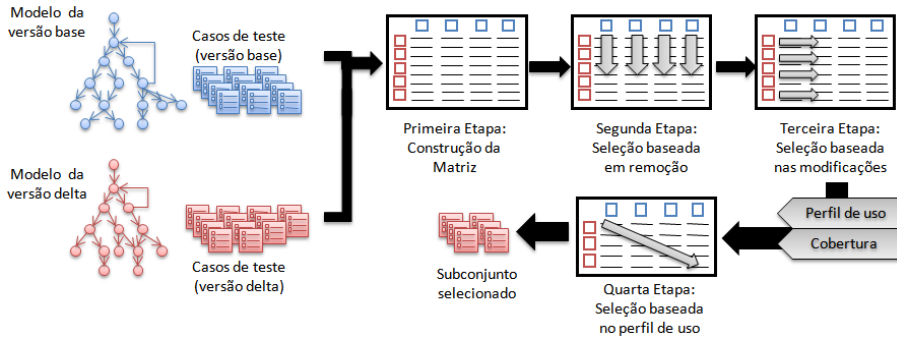


Figura 1. Etapas de execução de WSA-RT.

Em WSA-RT, T' apresenta as modificações realizadas no software (e.g. adição, remoção e mudanças nas transições da aplicação). Por sua vez, o modelo da versão delta, assim como os casos de teste gerados a partir dele, possuem as informações do perfil de uso fornecido pelo testador uma vez que este modelo se refere à versão atual do sistema.

A técnica utiliza como entrada o perfil de uso do modelo modificado e os dois conjuntos de casos de teste (T e T'). O conjunto selecionado é subconjunto de T' , de forma que este subconjunto não apresenta casos de teste obsoletos, e apresenta casos de teste que interagem com as modificações no modelo [4]. Ao selecionar um subconjunto de casos de teste que interagem com as modificações realizadas, é possível aumentar as chances de revelar os defeitos de regressão. Para ilustrar as etapas e os passos da técnica, será utilizado um exemplo descrito na próxima subseção. As outras seções descrevem as etapas da técnica.

3.1 Exemplo de Versões Base e Delta

O sistema de transições rotuladas (STR) da Figura 2 representa o modelo comportamental do sistema que desejamos testar. Nesta figura, são especificados o modelo da versão base (a) e o modelo da versão delta (b) com as respectivas probabilidades do perfil de uso.

As transições do modelo da versão delta apresentam pesos, indicando a probabilidade de execução da respectiva transição (e.g. um passo da aplicação, ou do caso de uso). Em WSA-RT, assim como no trabalho proposto em WSA, a soma dos pesos associados às transições saindo de um mesmo estado deve *somar* 1. Esta característica pode ser observada nos estados 1, 2, 5 e 7 da Figura 2 (b).

No modelo delta do exemplo ilustrado, ao atingir o estado 5, o perfil especificado

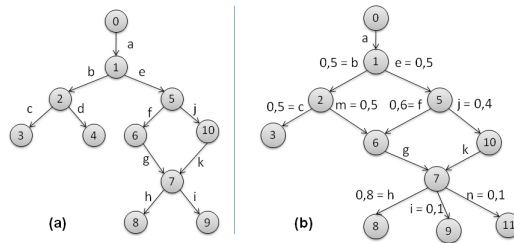


Figura 2. Modelos – base (a) e delta com os pesos (b) – utilizados para ilustrar WSA-RT.

indica que há uma probabilidade de 60% de executar a transição “f” e 40% de executar a transição “j”. A probabilidade das transições cujo estado de origem não possui outras transições – e.g. os estados 0, 6 e 10 da Figura 2 (b) – é considerado 1.

Apesar dos pesos em WSA-RT representarem as probabilidades de execução das transições, os pesos podem representar outro aspecto da aplicação. Um exemplo seria a probabilidade da respectiva transição revelar um defeito. Devido à subjetividade relacionada com os pesos especificados pelo testador, não foi possível desenvolver diretrizes para facilitar a atribuição de pesos. Dessa forma, esta etapa ainda é dependente da experiência e habilidade do testador em atribuir bons pesos às transições (e.g. valores bem distribuídos capazes de refletir a importância dos cenários).

Além dos pesos, é possível observar que nesse exemplo foram realizadas apenas 3 modificações nas transições do modelo: a remoção da transição de rótulo “d” (e consequentemente do estado 4), a adição da transição “m” entre os estados 2 e 6; e a adição da transição “n” no estado 7 (e consequentemente a adição do estado 11). A partir dos modelos do exemplo, foram gerados os casos de teste (caminhos no modelo) apresentados na Tabela 1. Esta tabela apresenta os casos de teste de T e de T' , assim como o valor de probabilidade de cada caso de teste de T' (apresentado na última coluna). Este valor representa a probabilidade de que os passos exercitados pelo caso de teste sejam executados pelo usuário da aplicação. Esta probabilidade é obtida multiplicando o peso de cada transição do caso de teste. O exemplo descrito nesta subseção será utilizado para ilustrar as etapas da técnica.

3.2 Construção da Matriz de Similaridade

A primeira etapa da técnica é a construção da matriz de similaridade – Figura 1 (a). Esta matriz apresenta a similaridade entre pares de casos de teste provenientes de dois conjuntos de casos de teste diferentes. Apesar de diferentes, é desejável que estes conjuntos tenham sido gerados a partir de duas versões distintas do mesmo software, pois os trechos

Tabela 1. Casos de testes gerados automaticamente a partir dos modelos.

Versão base				Versão delta				Probabilidade de execução
TC_1	a	b	c	TC'_1	a	b	c	0,25
TC_2	a	b	d	TC'_2	a	b	m	0,2
TC_3	a	e	f	TC'_3	a	b	m	0,025
TC_4	a	e	f	TC'_4	a	b	m	0,025
TC_5	a	e	j	TC'_5	a	e	f	0,24
TC_6	a	e	j	TC'_6	a	e	f	0,03
				TC'_7	a	e	f	0,03
				TC'_8	a	e	j	0,16
				TC'_9	a	e	j	0,02
				TC'_{10}	a	e	j	0,02

em comum entre as duas versões (i.e. os trechos não modificados) influenciam no cálculo de valor de similaridade realizado pela técnica.

Em WSA-RT, cada linha da matriz representa um caso de teste de T' , enquanto que cada coluna representa um caso de teste de T . Portanto, a informação de similaridade entre cada par de caso de teste (TC'_i, TC_j), onde $TC'_i \in T'$, e $TC_j \in T$, representa a similaridade entre os casos de teste das versões base e delta (colunas e linhas, respectivamente). Dessa forma, a matriz de similaridade $A_{|T'| \times |T|}$ é calculada da seguinte forma:

$$(a) a[i, j] = \frac{QtdPassosIguais(TC'_i, TC_j)}{TamanhoMédio(TC'_i, TC_j)} \quad (b) TamanhoMédio(TC'_i, TC_j) = \frac{|TC'_i| + |TC_j|}{2} \quad (1)$$

As equações 1(a) e 1(b) foram propostas por Cartaxo et al. [8] e adaptadas neste trabalho para o contexto de teste de regressão. A função $QtdPassosIguais(TC'_i, TC_j)$ no numerador da Equação 1(a) calcula a quantidade de passos iguais entre os casos de testes especificados como parâmetros, onde dois passos (ou transições) são iguais, em um modelo, quando possuem os mesmos: vértice de *origem*, *rótulo* de transição e vértice de *destino*.

O denominador da Equação 1(a) equilibra a quantidade de passos iguais pela média aritmética entre o tamanho de cada caso de teste do par analisado. Esta divisão é realizada com o objetivo de normalizar os valores de similaridade com relação aos diversos tamanhos dos casos de testes. Neste trabalho, o tamanho de um caso de teste ($|TC_x|$) é a quantidade de passos (no nosso caso, transições no modelo) que este possui. Considerando os casos de teste TC'_3 e TC_5 , obtemos o tamanho médio 5 e o valor de similaridade 0,20.

Os valores de similaridade 1 são utilizados para identificar os casos de teste que não interagem com nenhuma modificação, ou seja, o mesmo cenário é exercitado nas duas versões. É possível observar esta característica da técnica, utilizando a Equação 1(a) para TC'_1

e TC_1 ($TC_1 = TC'_1 = [a, b, c]$). Na Tabela 2, é apresentada a matriz de similaridade, obtida a partir de T e T' .

Tabela 2. Matriz de similaridade obtida a partir dos CCTR do exemplo.

Casos de teste	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6
TC'_1	1,00	0,66	0,25	0,25	0,25	0,25
TC'_2	0,50	0,50	0,60	0,40	0,40	0,20
TC'_3	0,50	0,50	0,40	0,60	0,20	0,40
TC'_4	0,50	0,50	0,40	0,40	0,20	0,20
TC'_5	0,25	0,25	1,00	0,80	0,60	0,40
TC'_6	0,25	0,25	0,80	1,00	0,40	0,60
TC'_7	0,25	0,25	0,80	0,80	0,40	0,40
TC'_8	0,25	0,25	0,60	0,40	1,00	0,80
TC'_9	0,25	0,25	0,40	0,60	0,80	1,00
TC'_{10}	0,25	0,25	0,40	0,40	0,80	0,80

Observando os maiores valores de similaridade em cada linha e coluna, é possível encontrar os casos de teste modificados e os obsoletos, como ilustrado na Tabela 3. Além disto, os valores de similaridade na matriz são utilizados para identificar os casos de teste da versão delta que são mais similares aos da versão base. Os casos de teste *menos similares* entre si exercitam *mais modificações*, uma vez que os CCTR são gerados a partir de versões diferentes de uma mesma aplicação.

Tabela 3. Significado dos maiores valores de similaridade das linhas e colunas.

Maior similaridade:	Na Coluna	Na Linha
é diferente de 1	caso de teste é obsoleto	caso de teste exercita adições ou modificações no modelo
é igual a 1	caso de teste que não interage com nenhuma modificação	

3.3 Seleção Baseada em Casos de Teste Obsoletos

Em WSA-RT, um caso de teste obsoleto é identificado quando não é encontrado um valor de similaridade 1 na sua respectiva coluna. Ao identificar que a coluna corresponde a um caso de teste obsoleto, selecionamos o caso de teste da versão delta *mais similar* a ele. Por meio desta estratégia de seleção, o caso de teste selecionado exercita a maior quantidade de transições presentes no caso de teste obsoleto aumentando, dessa forma, a probabilidade de encontrar algum defeito de regressão devido à remoção [3]. Se, ao analisar uma coluna, for encontrado um empate entre duas ou mais linhas na matriz, a linha selecionada é escolhida aleatoriamente, dentre as empatadas.

Observando a matriz (Tabela 2), é possível verificar que a coluna referente ao caso de teste TC_2 é a única que não apresenta similaridade 1 nas suas linhas. Portanto, TC_2 é

um caso de teste obsoleto. Podemos observar que a identificação realizada por WSA-RT é correta, pois TC_2 exercita a única transição removida do modelo – transição “ d ” da Figura 2 (a). Iterando sobre cada linha da coluna de TC_2 , o maior valor de similaridade encontrado é 0,66. Diante disto, é selecionado o caso de teste TC'_1 da versão delta. Como as demais colunas apresentam o valor de similaridade 1, a técnica prossegue para a próxima etapa.

3.4 Seleção Baseada em Modificações

Após identificar as transições removidas, WSA-RT observa as adições e modificações das transições do modelo. O objetivo desta etapa é remover da matriz e, portanto, do subconjunto selecionado, os casos de teste que não interagem com nenhuma modificação realizada no modelo. Estes casos de teste não possuem uma alta probabilidade de revelar os defeitos de regressão [1], e podem ser identificados através de um valor de similaridade 1 na respectiva linha da matriz. Dessa forma, as linhas que apresentam um valor de similaridade igual a 1 são removidas da matriz, exceto por aquelas linhas que já foram selecionadas na etapa anterior da técnica (a seleção baseada nos casos de teste obsoletos – Seção 3.3).

Observando o exemplo da Tabela 2, é possível identificar que as linhas TC'_1 , TC'_5 , TC'_6 , TC'_8 e TC'_9 apresentam o valor de similaridade 1, quando comparados, respectivamente, aos casos de teste TC_1 , TC_3 , TC_4 , TC_5 e TC_6 da versão base. Portanto, estes casos de teste não interagem com adições no modelo e devem ser removidos da matriz.

No entanto, é importante observar que TC'_1 já foi selecionado, pois é o caso de teste mais semelhante a TC_2 da versão base. Portanto TC'_1 interage com uma modificação do modelo (uma remoção de transição) e não deve ser removido do subconjunto. Dessa forma, apenas os casos de teste TC'_5 , TC'_6 , TC'_8 e TC'_9 são removidos da matriz, e consequentemente do subconjunto.

3.5 Seleção Baseada no Perfil de Uso

Em algumas situações, é possível que a modificação seja tão severa que a quantidade de casos de teste que interagem com estas modificações seja proporcionalmente grande, de forma que os recursos disponíveis para a realização do teste de regressão não sejam suficientes. Diante disto, é necessário remover mais casos de teste do subconjunto. Sob esta perspectiva, WSA-RT inicia a execução do último passo, caracterizado pela *remoção dos casos de teste menos importantes, dentre os que interagem com as modificações*.

Para identificar os casos de teste menos importantes, os valores de similaridade de cada linha são divididos pela probabilidade do respectivo caso de teste da versão delta. Após dividir os valores da matriz de similaridades, o testador especifica uma porcentagem de cobertura, referente à quantidade de casos de teste desejada. Esta porcentagem deve refletir a

quantidade de casos de teste que podem ser executados de acordo com a disponibilidade dos recursos para realizar o teste de regressão. Diante disto, a técnica realiza várias buscas dentre os valores da matriz para remover o caso de teste correspondente à linha em que o maior valor da matriz é encontrado. De acordo com WSA-RT, esse caso de teste é o menos importante dentre os que interagem com as modificações.

Até então, no exemplo ilustrado, os casos de teste selecionados foram: TC'_1 , TC'_2 , TC'_3 , TC'_4 , TC'_7 e TC'_{10} . A Tabela 4 apresenta a matriz do exemplo, após as divisões pelos valores de probabilidade dos casos de teste.

Tabela 4. Matriz de similaridade após a divisão pelas probabilidade.

Casos de teste	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6
TC'_1	4	2,6	1	1	1	1
TC'_2	2,5	2,5	3	2	2	1
TC'_3	20	20	16	24	8	16
TC'_4	20	20	16	16	8	8
TC'_7	8,3	8,3	26,6	26,6	13,3	13,3
TC'_{10}	12,5	12,5	20	20	40	40

A partir da matriz da Tabela 4 e considerando, por exemplo, uma cobertura de 50% dos caso de teste do conjunto obtido até então, é desejado um subconjunto com apenas 3 casos de teste. Durante a primeira busca na matriz, o maior valor encontrado é 40, na última linha da matriz. Dessa forma, o TC'_{10} é removido do conjunto e da matriz. Pela cobertura especificada, outros dois casos de teste devem ser removidos. Observando a matriz são encontrados os valores 26,6 e 24, o que implica na remoção de TC'_7 e TC'_3 , respectivamente.

Após estas remoções, a cobertura especificada é alcançada e a técnica encerra a execução. O subconjunto obtido apresenta os casos de teste TC'_1 , TC'_2 e TC'_4 . Como é possível observar na Tabela 5, o CCTR resultante cobre as modificações realizadas e apresenta os maiores valores de probabilidade especificados.

Tabela 5. Subconjunto selecionado por WSA-RT.

	Casos de Teste					Probabilidade de execução
TC'_1	a	b	c			0,25
TC'_2	a	b	m	g	h	0,2
TC'_4	a	b	m	g	n	0,025

4 Estudos de caso

Uma avaliação de WSA-RT foi realizada a partir de três estudos de caso³, comparado-a com a técnica proposta por Chen et al. [5] que é a mais diretamente relacionada com WSA-RT presente na literatura. Esta técnica também utiliza uma especificação de valores (no caso, valores de riscos) fornecida pelo testador e observa modificações realizadas no modelo a fim de encontrar casos de teste que exercitam os trechos modificados, bem como casos de teste que cobrem requisitos de interesse do testador. O desempenho é observado verificando o potencial de redução de cada técnica, assim como, a densidade de falhas⁴ no CCTR selecionado. Considerando T (o CCTR), T' (os casos de teste selecionados pela técnica), $\overline{T'}$ (os casos de teste removidos pela técnica), e T_F (os casos de teste que falham devido ao(s) defeito(s) de regressão), temos que as métricas são observadas da seguinte forma:

$$(a) \text{ Potencial de Redução} = 100 \times \frac{|T'|}{|T|}. \quad (b) \text{ Densidade de Falhas} = 100 \times \frac{|T_F|}{|T'|}. \quad (2)$$

Utilizando estas métricas, é possível identificar a técnica capaz de reduzir mais o CCTR e também revelar os defeitos de regressão. A seguir, são discutidos a configuração, os resultados e as ameaças à validade do estudo de caso.

4.1 Configuração dos Estudos de Caso

Para a realização dos estudos de caso, foram utilizadas especificações das ferramentas *Labeled Transitions System - Based Testing* (LTS-BT), *Test Case Selector* (TCS) e PDFSam. LTS-BT e TCS são ferramentas que realizam geração e seleção automática de casos de teste [11, 12]. Por sua vez, PDFSam é uma ferramenta desenvolvida para manipular (e.g. visualizar, cortar, combinar) arquivos no formato PDF (*Portable Document Format*). A especificação de cada ferramenta consiste em: um documento de casos de uso (no formato proposto por Cabral e Sampaio [13]); e um diagrama de atividades UML.

Para realizar o re-teste seletivo, foram realizadas modificações na especificação de cada ferramenta, que refletiam na adição e remoção de transições do seu respectivo modelo. Também foram estabelecidos os defeitos de regressão, associados às diversas modificações realizadas. As modificações e defeitos de regressão foram estabelecidos pro meio da análise dos documentos de modificações (*changelogs*) e correções nas versões de cada ferramenta. Além disto, um testador experiente com técnicas baseadas em modelo que atribuiu os valores de probabilidades analisados por WSA-RT. Para a técnica de Chen et al., foi solicitado a

³Os artefatos destes estudos de caso estão em: <http://tinyurl.com/3p3gf7g>

⁴Neste trabalho, as falhas indicam casos de teste que falham devido a pelo menos um defeito de regressão.

especificação dos valores de riscos analisados pela técnica, de acordo com as diretrizes propostas pelos autores da técnica [5]. Após gerar manualmente os modelos das duas versões, as modificações, as falhas de regressão e o perfil de uso, foi iniciada a execução do estudo de caso. Os tamanhos de cada CCTR (gerado automaticamente a partir do modelo de especificação), e os demais detalhes de configurações de cada estudo de caso são apresentados na Tabela 6.

Tabela 6. Detalhes sobre os estudos de caso.

	Aplicação	Modificações	Defeitos (Regressão)	Transições	Tamanho CCTR	
					Base	Delta
Est. de Caso 1	TCS	3	3	39	34	37
Est. de Caso 2	PDFSam	4	6	74	72	81
Est. de Caso 3	LTS-BT	6	10	126	55	65

4.2 Resultados e Análise

Uma vez gerados os CCTR, as técnicas foram executadas considerando uma seleção de 75% de casos de teste. Em ambas as técnicas, a seleção baseada em valores ocorre após a seleção dos casos de teste que interagem com as modificações, portanto, esta cobertura foi definida com o objetivo de não reduzir muito os conjuntos obtidos após a análise das modificações. Devido à presença de aleatoriedade em alguns trechos do algoritmo das técnicas, foram realizadas 100 execuções de cada técnica, e o resultado obtido foi a média aritmética destas execuções. Os resultados são apresentados na Tabela 7.

Tabela 7. Resultados do estudo de caso.

		Chen et al.	WSA-RT
Estudo de Caso 1	Potencial de Redução	26,47%	78,37%
	Densidade de Falhas	20%	87,5%
Estudo de Caso 2	Potencial de Redução	25%	74,07%
	Densidade de Falhas	44,44%	80,95%
Estudo de Caso 3	Potencial de Redução	10,90%	70,10%
	Densidade de Falhas	36,70%	62,02%

As duas técnicas foram capazes de cobrir todas as modificações, porém, não foram selecionados os mesmos subconjuntos. Como podemos ver na Tabela 7, WSA-RT apresentou um melhor desempenho tanto no potencial de redução como na densidade de falhas para todos os estudos de caso. O potencial de redução apresentado por WSA-RT foi muito maior do que a redução obtida por Chen et al., em todos os estudos de caso realizados. Este resultado é atribuído ao algoritmo de seleção baseada nas modificações realizado pelas duas técnicas.

Ao encontrar uma modificação, a técnica de Chen et al. seleciona casos de teste que exercitam o fluxo modificado, e *os demais fluxos* que o seguem. Como a maior parte dos cenários nos estudos de caso apresentam fluxos alternativos, a quantidade de casos de teste selecionados aumenta. Por outro lado, WSA-RT seleciona apenas o caso de teste que exercita o fluxo associado à modificação, diminuindo a quantidade de casos de teste de regressão selecionados e proporcionando, portanto, uma maior redução no CCTR.

Os resultados de densidade de falhas também seguem a mesma perspectiva. A grande quantidade de casos de teste selecionados pela técnica de Chen et al. diminuiu consideravelmente a densidade do subconjunto obtido, pois foram selecionados muitos casos de teste que não revelam defeitos de regressão. Por outro lado, WSA-RT foi capaz de remover aqueles que não revelam os defeitos. Durante a análise dos resultados do *Estudo de Caso 2*, observamos que a quantidade de caminhos com loops nos cenários do modelo contribuiu para um aumento na quantidade de casos de teste gerados, e portanto, um aumento significativo no potencial de redução das técnicas (em especial para WSA-RT) e na densidade de falhas obtida (cerca de 87%). Para a técnica Chen et al., o aumento na densidade foi significativo (cerca de 20% a mais que no *Estudo de Caso 1*), porém WSA-RT foi capaz de se beneficiar mais da redundância presente no CCTR (capturando cerca de 47% a mais de casos de testes que revelam defeitos de regressão). Este resultado indica que WSA-RT é capaz de lidar com as redundâncias presentes no CCTR, aumentando o potencial de redução, sem comprometer significativamente a capacidade de revelar defeitos de regressão.

Com relação à seleção baseada em valores, foi observado que ambas as técnicas selecionaram os 75% dos casos de teste que apresentavam os respectivos maiores valores. Ou seja, do CCTR obtido após a análise de modificações, cada técnica removeu 25% de casos de teste. Como a técnica de Chen et al. reduziu menos casos de teste associados às modificações, foi removida uma quantidade menor de casos de teste na análise baseada em valores. Porém, observando os resultados, foi possível concluir que esta etapa não contribuiu significativamente para o aumento na capacidade de revelar os defeitos de regressão. Contudo, selecionar os casos de teste importantes contribuem com a validação do software, uma vez que de acordo com a abordagem baseada em valores utilizada, os cenários exercitados por estes casos de teste são frequentemente executados pelo usuário.

O tempo de execução da seleção automática realizada pelas técnicas foi entre 10 e 50 milissegundos. Como os casos de testes gerados são abstratos, seria necessário muito tempo para a execução (manual) completa do CCTR, de forma que a utilização das técnicas possibilita um ganho no tempo de execução do teste de regressão. No caso de WSA-RT, reduzimos o CCTR em cerca de 70% sendo foi possível cobrir todas as modificações realizadas nos modelos. Se considerarmos que cada caso de teste requer o mesmo tempo de execução a redução no tempo de realização do teste é significativa, porém, o tempo de execução dos casos de teste nas respectivas ferramentas não foi disponibilizado para análise.

As principais ameaças à validade do estudo de caso são: a utilização de apenas uma especificação de valores em cada estudo de caso, e a utilização de poucas especificações. Ambas as ameaças comprometem o poder de generalização dos resultados obtidos, de forma que um estudo experimental considerando várias configurações de valores e diversos modelos (e suas respectivas modificações) seria capaz de prover informações mais precisas acerca do desempenho de WSA-RT.

5 Trabalhos Relacionados

Diversos trabalhos para o re-teste seletivo têm sido propostos na literatura. No entanto, grande parte destes são para o contexto de código [3]; ainda há vários problemas em aberto com relação ao re-teste seletivo baseado em especificação. Na maioria das técnicas baseadas em especificação, as modificações são identificadas comparando as versões diferentes dos modelos. Diante disto, vários modelos e as suas respectivas entidades (e.g.: fluxos, atividades, classes, estados, atributos, etc.) são investigadas automaticamente pelas técnicas. Geralmente, as técnicas diferem no critério utilizado para seleção; um exemplo seria a cobertura de: transições, estados, valores de riscos, defeitos estimados, dentre outros.

Korel *et al.* [3] propõem uma técnica, onde, a partir de uma Máquina de Estados Finita Estendida (MEFE), é realizada uma análise de dependência entre os elementos do modelo e as modificações realizadas. Esta análise permite identificar as modificações e as respectivas entidades afetadas por meio de conceitos de dominância entre as transições e estados da MEFE. Por estar relacionada com estruturas da MEFE, a utilização de outro modelo de entrada (e.g. diagramas UML) pode comprometer o desempenho e aplicabilidade da técnica. Além disto, a análise de dependência pode ser cara e impraticável em MEFES grandes.

A técnica de Análise de Riscos em Diagramas de Atividades proposta por Chen *et al.* [5] realiza uma seleção de casos de teste baseada nas modificações e na análise de riscos especificados pelo testador. No entanto, essa seleção baseada nas modificações reduz pouco o CCTR (Seção 4), devido à seleção de muitos casos de teste relacionados com poucas modificações. Isto indica que a análise de modificações realizada por WSA-RT é capaz de contemplar o mesmo conjunto de modificações porém selecionando menos casos de teste.

Naslavsky *et al.* [14] apresentam uma técnica de re-teste seletivo baseada em modelos, para realizar a seleção de casos de teste de regressão para testar a implementação, observando as entidades modificadas. Para isto, são utilizados diagramas de classes e diagramas de seqüências UML. Apesar de utilizar modelos de especificação, essa técnica ainda utiliza muitos recursos do código, dificultando a sua utilização para o teste de sistemas.

Por fim, diferentemente de outras técnicas, WSA-RT apresenta uma seleção que não utiliza um formato específico de modelo ou nível de abstração (código, ou sistema). Estas

características refletem a versatilidade de utilização da técnica para o teste de regressão.

6 Considerações Finais

O principal objetivo de WSA-RT é incorporar elementos de uma abordagem baseada em valores no processo de seleção dos casos de teste de regressão. Diante disto, WSA-RT é capaz de selecionar os casos de teste mais importantes, dentre os que interagem com as modificações realizadas. Ao analisar o desempenho de WSA-RT no estudo de caso, observamos uma redução significativa (acima de 70%) do CCTR, em todos os estudos de caso, de forma que, grande parte deste subconjunto é composto por casos de teste que revelam defeitos de regressão. LTS-BT [11] fornece o suporte ferramental para a execução da técnica, uma vez que tanto WSA-RT como os estudos de casos foram implementados na ferramenta.

Apesar das vantagens, foi possível observar que muitos casos de teste do subconjunto selecionado por WSA-RT cobriam os mesmos defeitos. Sob esta perspectiva, é necessário melhorar o algoritmo de seleção procurando aumentar a capacidade de detecção de defeitos no subconjunto selecionado. Além disto, os diversos tipos de software podem apresentar diferentes características nos modelos de especificação (e.g. modularidade, tamanho e quantidade dos caminhos, etc.). Estas características se refletem nos casos de teste, e portanto, na análise de similaridades de WSA-RT. Dessa forma, um estudo de caso futuro é verificar o comportamento da técnica ao ser aplicada em vários modelos distintos.

Também é necessário realizar estudos experimentais para obter uma análise acerca do comportamento da técnica quando aplicações reais e CCTR maiores (com mais funcionalidades) são utilizados. Por meio dos resultados do experimento, é possível realizar melhorias no algoritmo de seleção para reduzir mais a redundância do CCTR selecionado. Além disto, é importante identificar as diretrizes para guiar o testador na atribuição de pesos, de forma que a especificação das probabilidades aproveite melhor os recursos da técnica (e.g. cobertura do modelo, porcentagem desejada, dentre outros).

7 Agradecimentos

Este trabalho foi apoiado pela CAPES e pelo Instituto Nacional de Ciência e Tecnologia para Engenharia de Software (INES⁵), financiado pelo CNPq, processos 573964/2008-4, 560014/2010-4 e 484643/2011-8.

⁵www.ines.org.br

Referências

- [1] Robert V. Binder. Francisco Gomes Testing object-oriented systems: models, patterns, and tools. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [2] Mary Jean Harrold and Alessandro Orso. Retesting software during development and maintenance. In Proceedings of FoSM 2008, pages 99–108, Beijing, China, October 2008.
- [3] B. Korel, L. H. Tahat, and B. Vaysburg. Model based regression test reduction using dependence analysis. In Proceedings of ICSM 2002, pages 214–223, USA, 2002. IEEE.
- [4] Mahadevan Subramaniam, Ling Xiao, Bo Guo, and Zoltan Pap. An approach for test selection for EFSMs using a theorem prover. In Proceedings of TESTCOM '09/FATES '09, pages 146–162, Berlin, Heidelberg, 2009. Springer-Verlag.
- [5] Yanping Chen, Robert L. Probert, and D. Paul Sims. Specification-based regression test selection with risk analysis. In Proceedings of CASCON'02. IBM Press, 2002.
- [6] A. Bertolino, E. Cartaxo, P. D. L. Machado, and E. Marchetti. Weighting influence of user behavior in software validation. In Proceedings of the 19th DEXA 2008 Workshops, pages 495–500. IEEE Computer Society, 2008.
- [7] F. G. Oliveira Neto and P. D. L. Machado. WSA-RT: Uma técnica para a seleção de casos de teste de regressão baseados na especificação do sistema. In Proceedings of CBSOFT/SAST 2011, 2011.
- [8] Emanuela Gadelha Cartaxo, Patricia Duarte Lima Machado, and Francisco Gomes Oliveira Neto. On the use of a similarity function for test case selection in the context of model-based testing. STVR Journal of Software Testing, Verification, and Reliability, pages 75–100, 2009.
- [9] Hadi Hemmati, Andrea Arcuri, and Lionel Briand. Empirical investigation of the effects of test suite properties on similarity-based test case selection. In Proc. of the 4th ICST 2011, 2011.
- [10] Barry Boehm. *Value-Based Software Engineering: Overview and Agenda*. pages 3–14. Springer Verlag, 2006.
- [11] Emanuela G. Cartaxo, Wilkerson L. Andrade, Francisco Gomes Oliveira Neto, and Patrícia D. L. Machado. LTS-BT: a tool to generate and select functional test cases for embedded systems. In Proceedings of the ACM SAC 2008, pages 1540–1544, New York, NY, USA, 2008. ACM.

- [12] Juliana Mafra, Breno Miranda, Juliano Manabu Iyoda, and Augusto César Sampaio. Test case selector: Uma ferramenta para seleção de testes. In Proceedings of SBMF/SAST 2009, Porto Alegre, 2009. SBC.
- [13] Gustavo Cabral and Augusto Sampaio. Formal specification generation from requirement documents. *Electron. Notes Theor. Comput. Sci.*, 195:171–188, 2008.
- [14] Leila Naslavsky, Hadar Ziv, and Debra J. Richardson. A model-based regression test selection technique. In *ICSM*, pages 515–518, 2009.