

INTEGRAÇÃO DE MÉTODOS DE TESTE À FERRAMENTA WEB-PERFORMCHARTS

Daniele Fernandes Ferreira¹, Mirian Karime Pieve Nunes¹, Érica Ferreira², Alessandro Arantes², Nandamudi Lankalapalli Vijaykumar²

¹Faculdade de Tecnologia de São José dos Campos - SP

²Instituto Nacional de Pesquisas Espaciais, INPE - São José dos Campos, SP

daniele_ff@yahoo.com.br, ericaferrso@gmail.com

1. Introdução

A *WEB-PerformCharts* é uma ferramenta de teste de software desenvolvida pelo Instituto Nacional de Pesquisas Espaciais (INPE) que possibilita a geração de casos de testes através da Internet. Esta ferramenta converte uma especificação em *Statecharts* [3], que é uma representação sintética de sistemas críticos e complexos, em uma Máquina de Estados Finitos (MEF) [1].

Os modelos de teste gerados a partir de sistemas críticos e complexos são amplos, o que complica o entendimento dos testadores. A técnica de *Statecharts* possibilita uma representação macro desses modelos, facilitando então o trabalho dos testadores. A ferramenta em questão foi criada de forma a automatizar os testes podendo realizá-los remotamente.

Atualmente a *WEB-PerformCharts* realiza testes com base nos métodos *Transition Tour* (TT) e *Switch Cover* [1]. No entanto, os pesquisadores do INPE acreditam que os métodos *Unique Input/Output* (UIO) e *Distinguishing Sequence* (DS) [2] também devem ser integrados na ferramenta.

Diante deste contexto, este trabalho tem como objetivo integrar os métodos DS e UIO à ferramenta *WEB-PerformCharts*.

2. Metodologia

Os métodos UIO e DS, desenvolvidos para gerar casos de teste a partir de modelos em *Statecharts*, adotam a mesma metodologia de funcionamento. Estes métodos lêem uma MEF escrita em XML, que está em um arquivo comum ao invés de uma base de dados, e geram os casos de teste apenas apresentado os mesmos na tela. Consequentemente, neste processo os dados não são armazenados, e com isso, pode haver perda de informação.

Para que não ocorra tal problema, é necessária a conexão com um banco de dados para que seja possível o armazenamento da MEF e dos casos de teste gerados pelos métodos DS e UIO. Esta base de dados a ser utilizada já foi desenvolvida, visto que os métodos TT e *Switch Cover* já armazenam informações em um banco por meio da ferramenta *WEB-PerformCharts*. Trata-se de duas bases de dados, uma para armazenamento dos dados de entrada (dados da MEF) e outra para o armazenamento dos dados de saída (casos de teste).

A adaptação dos métodos é feita criando a conexão com os dois bancos de dados para a busca e armazenamento de informações. Além disso, a adequação da ferramenta *WEB-PerformCharts* se faz

necessária para que os métodos UIO e DS também sejam acessíveis pela sua interface.

3. Resultados

Como já mencionado, a adaptação dos métodos UIO e DS é possível conectando-se a duas bases de dados. Feitas estas conexões, os métodos de teste buscam, interpretam os dados da MEF, e geram os casos de teste armazenando-os na base de dados de saída, como mostra a Figura 1.

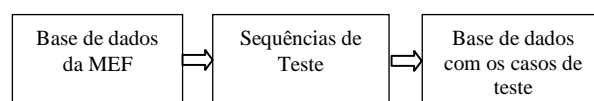


Figura 1 – Metodologia da geração de casos de Teste

A ferramenta de teste de software em questão (*WEB-PerformCharts*) foi desenvolvida na linguagem PHP e os métodos na linguagem JAVA. Ambos devem sofrer modificações em sua implementação.

O problema de perda de dados é extinto, e então, é realizado o armazenamento dos casos de teste gerados na base de dados. Tais alterações estão sendo feitas de modo a não comprometerem as implementações anteriores.

4. Conclusões

A integração de métodos de teste à ferramenta *WEB-PerformCharts* possibilita a geração dos casos de teste de maneira automatizada.

A importância da incorporação dos métodos UIO e DS à ferramenta atende a necessidade de se realizar a atividade de teste usando especificações em *Statecharts*, em sistema crítico e complexo.

5. Referências

- [1] Arantes, Alessandro Oliveira, *WEB-PerformCharts: A web-based Test Case Generator From Statecharts Modeling*, São José dos Campos: INPE, 2008.
- [2] Ferreira, E. , *Geração de casos de teste para sistemas da Área Espacial usando Critérios de Teste para Máquinas de Estados Finitos*. São José dos Campos: INPE, 2010.
- [3] Sidhu, Leung, T. , *Formal methods for protocol testing: A detailed study*. Transactions on Software Engineering, 1989.