

INVESTIGAÇÃO DE EVENTOS TEMPORIZADOS NA ESPECIFICAÇÃO FORMAL DE SISTEMAS REATIVOS PARA GERAÇÃO AUTOMÁTICA DE TESTES DE SOFTWARE

MIRIELE DOS S. SILVÉRIO¹, LUCIANA B. R. DOS SANTOS²

¹ Graduando em Tecnologia de Análise e Desenvolvimento de Sistemas, Bolsista PIBIFSP, IFSP, Câmpus Jacareí, mirielesilverio@hotmail.com

² Doutora em Computação Aplicada, Orientadora, Professora, IFSP, Câmpus Jacareí, lurebelo@ifsp.edu.br

RESUMO: Uma das principais métricas para aferir qualidade é a quantidade de defeitos encontrados nos sistemas de software. Muitos estudos têm sido desenvolvidos com o intuito de detectar defeitos nos sistemas, sendo que um dos mais usados para garantir a corretude de um produto de software, é o processo de teste. Quando se trata de teste de software, uma abordagem que se tornou popular é a de Testes Baseados em Modelo, onde a especificação do software é representada por um modelo. O modelo mais comum, devido à semelhança com os grafos, são as Máquinas de Estados Finitos (MEF), onde há um conjunto de estados e arcos de transição entre estes estados. A transição entre estados acontece devido à ocorrência de um evento, o qual pode estar relacionado por exemplo, ao término de um processamento, ao recebimento de uma ligação ou a um temporizador. Dessa forma, esse trabalho busca dar uma contribuição para melhorar a qualidade de software, utilizando para tal, eventos temporizados, onde a mudança de estado está relacionada com a passagem de tempo. Para isso, será proposta e implementada uma abordagem dentro da ferramenta Web-Performcharts que viabilize a utilização de eventos temporizados para geração de casos de teste.

PALAVRAS-CHAVE: Teste de Software; Testes Baseados em Modelo; Máquinas de Estados Finitos; Eventos temporizados

INVESTIGATION OF TIMED EVENTS IN SPECIFICATION REACTIVE SYSTEMS FORMAL FOR AUTOMATIC SOFTWARE TEST GENERATION

ABSTRACT: One of the main metrics for measuring quality is the number of defects detected in software systems. Many works have been developed in order to detect defects in the systems, one of the most used to guarantee the correctness of a software product, is the test process. Relating to software testing, Model-Based Testing is very popular, where the software specification is represented as a model. The most common model, due to the similarity with the graphs, are the Finite State Machines (FSM), where there is a set of states and transition arcs between these states. The transition between states occurs due to the occurrence of an event, which can be related, for example, to the end of a processing, to the receipt of a call or to a timer. Thus, this work seeks to make a contribution to improve the quality of software, using timed events, where the change of state is related to the passage of time. For that, an approach will be proposed and implemented within the Web-Performcharts tool that makes possible the use of timed events to generate test cases.

KEYWORDS: Software Testing; Model-based Tests; Finite State Machines; Timed Events.

INTRODUÇÃO

Nos dias atuais a presença de sistemas de software se dá de forma notável nas mais diferentes áreas. A construção dessas importantes ferramentas envolvem diversas fases, dentre elas a destacada nesse projeto é a de Verificação e Validação. Essa fase tem como fito assegurar que o software esteja em conformidade com sua especificação. Para o estabelecimento da confiança de que este está

adequado ao seu propósito diversas técnicas podem ser usadas, dentre elas a técnica de testes. Teste de software pode ser definido como:

O processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado. (NETO, 2009, p.54).

Os testes ainda podem ser de vários tipos, sendo um deles o teste formal. Esse tipo de teste permite verificar de forma automática se o produto desenvolvido tem correspondência com sua especificação formal. Dessa forma, dado um conjunto de especificações formais, esse método gera casos de testes, a fim de assegurar que a implementação atendeu as especificações. Para o levantamento de especificações formais, métodos também formais devem ser utilizados. Um método notável para essa listagem é o de utilização de modelos, esse trabalho irá abordar mais especificamente a utilização de Statecharts.

Quando erros são identificados nas etapas finais de desenvolvimento podem gerar altos custos, sobretudo em sistemas críticos, dessa forma, os processos aqui descritos contribuem positivamente para a identificação de problemas, e consequentemente na redução de custos de implementação.

Nesse contexto, o INPE - Instituto Nacional de Pesquisas Espaciais, construiu a ferramenta *PerformCharts*, cujo objetivo principal é a geração de casos de teste. No entanto, a plataforma arquitetada não compreende artefatos para a descrição e processamento de eventos temporizados, dessa forma, este trabalho se preconiza a sugerir e implementar no *PerformCharts* uma modelagem que descreva esses eventos.

SOLUÇÃO PROPOSTA

Esse trabalho encontra-se no contexto de uma ferramenta já existente, o *PerformCharts*.

O *PerformCharts* é uma ferramenta desenvolvida pelo INPE - Instituto Nacional de Pesquisas Espaciais, com a qual é possível gerar casos de teste a partir de diagramas *StateCharts*. A ideia inicial do projeto é que ele fosse capaz de avaliar sistemas reativos utilizando cadeias de Markov (ARANTES et al., 2002). A geração de casos de teste é uma das principais funcionalidades. Nesse caso, os modelos de especificação são convertidos para MEFs, e a partir delas as sequências de testes são geradas.

A ferramenta não disponibiliza uma interface gráfica para a modelagem dos *StateCharts*, portanto as especificações são dadas por meio do PcML (PerformCharts Markup Language)(AMARAL et al., 2003), uma linguagem de marcação baseada no XML.

No entanto, a marcação PcML não possui *tags* para a indicação de eventos temporizados. Sendo assim, esse projeto, com base nas especificações dadas pela W3C, propõe a seguinte marcação: Na transição entre eventos deve ser adicionado um atributo “Cond” o qual marcará a condicional relacionada ao temporizador. Para exemplificar, pense no caso de um microondas, o qual está modelado em *StateCharts* na Figura 1.

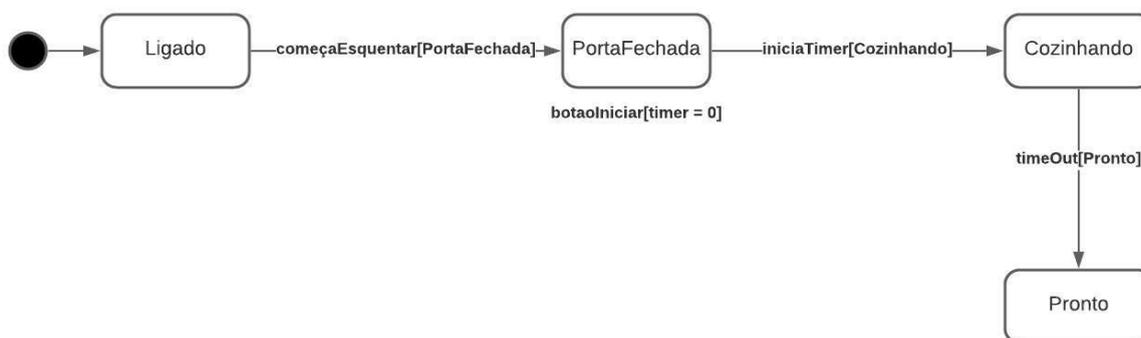


FIGURA 1: *StateCharts* microondas

Na Figura 1, é possível perceber que a transição do estado Cozinhando para Pronto depende de um evento temporizado, o *timeOut*. Segundo a modelagem PcML proposta, esse trecho deveria ser descrito da seguinte forma:

```
<Transition Cond="timer &gt;= cook_time" Destination="Pronto" Event="timeOut"/>
```

A modelagem PcML será então analisada pela ferramenta e convertida em uma estrutura de dados capaz de ser entendida pelo *PerformCharts* através de um processo descrito na Figura 2.

Esse processamento é feito através de um algoritmo implementado utilizando a linguagem de programação PHP, onde passará por duas fases principais:

- *Lexing*: Um analisador irá ler o arquivo, onde procurará *tokens* (expressões significativas/ reservadas, como é o caso do atributo proposto, Cond), e a partir disso entenderá o significado da estrutura descrita pelo PcML
- *Parsing*: O algoritmo de parseamento, após entender o que foi descrito em PcML, converterá os dados em uma estrutura capaz de ser interpretada e utilizada pelo *PerformCharts*, e então, armazenará essa estrutura em um banco de dados relacional, o qual atualmente utiliza o SGBD (Sistema Gerenciador de Banco de Dados) MySQL.

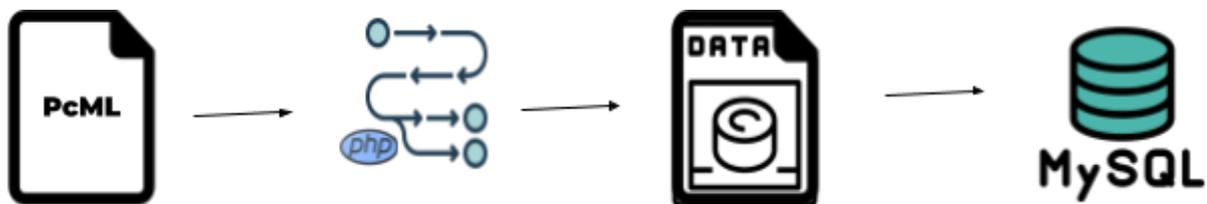


FIGURA 2: Modelagem do funcionamento do analisador de PcML

CONCLUSÕES

Pretende-se como trabalho futuro a consolidação do modelo PcML proposto, através da construção de análises léxicas adicionais na ferramenta *PerformCharts*. Dessa forma, a plataforma será capaz de disponibilizar os dados necessários para a construção de MEFs que incluam eventos temporizados, a fim de que casos de teste sejam gerados.

AGRADECIMENTOS

Ao IFSP, pela disponibilização da infraestrutura para realização desta pesquisa, e pelo apoio financeiro por meio da bolsa de Iniciação Científica.

REFERÊNCIAS

AMARAL, Ana Silvia M. S.; VIJAYKUMAR, N. L.; MARTINS, Eliane. Geração Automática de Casos de Teste de Conformidade para Software de Aplicações em Protocolos de Comunicação. [S. l.: s. n.], 2003. Disponível em: <http://mtc-m16c.sid.inpe.br/col/lac.inpe.br/worcap/2003/10.29.11.18/doc/final112003.pdf>.

Acesso em: 2 jul. 2020.

ARANTES, A. O.; VIJAYKUMAR, N. L.; SANTIAGO, V.; GUIMARAES, D. Test case generation for critical systems through a collaborative web-based tool. In: INTERNATIONAL CONFERENCE ON INOVATION IN SOFTWARE ENGINEERING (ISE 2008), 2008, Viena, Austria. ' Proceedings... Viena: IEEE Computer Society, 2002. p. 163–168

NETO, Arilo Cláudio Dias. Introdução a Teste de Software. Engenharia de Software Magazine , [s. l.], ed. 1, p. 54-59, 2009. Disponível em: https://www.researchgate.net/profile/Arilo_Neto/publication/266356473_Introducao_a_Testes

[_de_Software/links/5554ee6408ae6fd2d821ba3a/Introducao-a-Teste-de-Software.pdf](#). Acesso em: 1 jul. 2020