

## RESUMO

### CONVERSION OF UML ACTIVITY DIAGRAM FOR SAN: GENERATION OF SOFTWARE TEST CASES

- Orientador: Paulo Henrique Lemelle Fernandes
- Orientando: Toni Amorim<sup>32</sup>
- Instituição: Pontifícia Universidade Católica do Rio Grande do Sul
- Ano: 2010

**Abstract**  $\emptyset$  *Abstract - Recent Research suggest the use of formalisms such as stochastic Petri nets (SPN - Stochastic Petri Nets), Markov chains (MC - Markov Chains) and stochastic automata networks (SAN - Stochastic Automata Networks) to build a behavioral model which facilitates the analysis and validation of system during development, whose results make possible to develop a more robust and efficient software. Aiming to obtain a behavioral model of computing systems in this paper demonstrates a method of converting activity diagrams to SAN, a formalism that allows modeling of systems, from which it is possible to extract probabilistic indices related to states the model and generate test cases for software.*

**Index Terms**  $\emptyset$  (SAN - Stochastic Automata Networks), Software Testing, UML (Unified Modeling Language)

## INTRODUÇÃO

Apresentamos neste trabalho um método de conversão de diagramas de atividades para SAN fundamentado em um formalismo matemático que possibilita a modelagem de sistemas em geral, a partir do modelo comportamental de sistema construído por diagramas de atividades UML e com base dos modelos criados, é possível extrair índices probabilísticos relacionados aos estados do modelo SAN gerado, permitindo a geração de casos de testes de software cujos resultados podem fundamentar decisões de projeto.

O desenvolvimento de software envolve tanto a modelagem quanto o uso de linguagens orientadas a objeto, buscando empregar sempre os elementos da Engenharia de Software, dentre eles a UML (Unified Modeling Language) [2].

A UML foi desenvolvida com o objetivo de descrever qualquer tipo de sistema, em termos de diagramas orientados a objetos e de padronizar a forma de especificação de sistemas, desde as fases iniciais até a fase de testes e manutenção.

Durante o processo de desenvolvimento de software, há diversas atividades que tem por objetivo garantir a qualidade do produto a ser entregue ao cliente. Porém erros ainda podem aparecer, de modo que os testes são executados para garantir a identificação e resolução de possíveis erros.

Testes de software podem ser vistos como o processo de executar ações visando identificar a presença de erros ou limitações de desempenho do sistema, ou seja, consiste na verificação dinâmica do funcionamento de um determinado

---

<sup>32</sup> Toni Amorim, Professor Assistente Mestre da Universidade do Estado de Mato Grosso – UNEMAT – Campus Regional de Alto Araguaia, Rua Santa Rita, 148 Alto Araguaia, MT, Brazil, toniamorim@gmail.com

programa, baseado em um conjunto finito de casos de testes, intimamente relacionados dentro de um domínio infinito de entradas possíveis contra o funcionamento esperado [3].

Redes de Autômatos Estocásticos é um formalismo proposto por Plateau [4] nos anos 80 para modelagem analítica de sistemas com um grande espaço de estados representado por meio de subsistemas equivalente a cadeia de Markov [1].

Uma SAN é composta por um conjunto finito de estados e um conjunto finito de transições entre os estados, as quais são disparadas por eventos que possuem uma taxa de ocorrência associada a ele. O estado de um autômato representa toda a informação referente a seu passado, de modo que para um determinado conjunto de estados um sistema poderá assumir somente um estado a cada momento, sendo este denominado como estado local [5].

Os eventos sincronizantes disparam transições que mudam o estado local de mais de um autômato, isto é, dois ou mais autômatos podem mudar seus estados locais simultaneamente, naturalmente ocasionando também uma mudança no estado global da SAN.

## MÉTODO DE CONVERSÃO E A GERAÇÃO DOS CASOS DE TESTE

O método apresentado neste trabalho é uma conversão manual de diagramas de atividades da UML para um modelo SAN que possibilite realizar análises de comportamento, tais como, determinar a funcionalidade do sistema que possui uma maior probabilidade de uso. Com base no modelo SAN obtido é possível gerar casos de teste de software. O processo de conversão pode ser descrito em sete passos.

## RESULTADOS

Os resultados obtidos a partir do modelo SAN gerado demonstram a possibilidade de modelo SAN estar em um determinado estado que representa a execução de uma atividade.

A partir do modelo SAN obtido, foram gerados casos de teste de software onde cada caminho percorrido nos estados existentes representa um caso de teste.

## CONCLUSÃO

Este trabalho teve por objetivo apresentar um método de conversão de diagramas de atividades da UML para SAN e a geração de casos de teste de software.

Foram apresentados os principais conceitos sobre UML, a descrição das principais técnicas de teste de software, tais como teste estrutural e teste funcional, bem como os conceitos sobre o formalismo

SAN, buscando relacionar em bom nível de detalhamento as principais contribuições desses temas para o trabalho aqui apresentado.

Trabalhos futuros deverão abordar a melhoria no método de modo a reduzir o estado global dos modelos SAN gerados.

Ainda como trabalho futuro, espera-se encontrar outras formas de efetuar o tratamento das condições de guarda, visando obter modelos SAN com um tamanho menor, o que possibilitaria analisar sistemas mais complexos.

#### Referências Bibliográficas

- [1] G. Florin and S. Natkin. Les re´seaux de Petri stochastiques. *Tecniques et Sciences Informatiques*, 4(1):143–160, 1985.
- [2] Sommerville. I. *Engenharia de Software*. Pearson Addison Wesley, São Paulo, SP, 2007.
- [3] PEZZÉ E.M. and YOUNG M. *Teste e análise de software: processo, princípios e técnicas*. Bookman, Porto Alegre RS, 2008.
- [4] B. Plateau and K. Atif. Stochastic Automata Networks for modelling parallel systems. *IEEE Transactions on Software Engineering*, 17(10):1093–1108, October 1991.
- [5] A. Sales. *Formalismos Estruturados de Modelagem para Sistemas Markovianos Complexos*. Master’s thesis, PUCRS-FACIN-PPGCC, Porto Alegre, December 2003.