



Ferramenta de Modelagem e Geração de Testes Comportamentais de Software Embarcado

Thiago Campos Teixeira e Nascimento, Lucas Campos Teixeira e Nascimento, Rogério Atem de Carvalho

Para suprir a demanda cada vez maior por um desenvolvimento mais produtivo de sistemas embarcados, o emprego de múltiplos conceitos de desenvolvimento ágil de software em projetos de sistema de informação é amplamente incentivado. Devido à natureza dos sistemas embarcados, há uma grande vantagem em ter como foco seu comportamento durante as fases de identificação e projeto. Se beneficiando dessas vantagens, o emprego do método de modelagem WOOM (do inglês Workflow, Object-Oriented Method) se mostra extremamente valioso pois ele permite a modelagem da estrutura e comportamento de um dado sistema utilizando conceitos estabelecidos de UML e Contratos. Somando esta decisão de foco à aplicação de técnicas ágeis com o método de modelagem, é possível elaborar um projeto em que a perda de requisitos e informações estruturais do sistema são evitadas, também possibilitando realizar testes simultaneamente à etapa de desenvolvimento. Com o objetivo de auxiliar o desenvolvedor do sistema a ser elaborado de forma que o mesmo se beneficie com essas vantagens, desenvolveu-se uma extensão para o software de modelagem *UML*, *Modelio*, que faz uso das ferramentas já existentes deste programa e o expande com a adição de dispositivos prontos para a utilização do usuário. A extensão foi desenvolvida em Python e é executada por meio do sistema de scripts do *Modelio*. Ela interage diretamente com a funcionalidade de geração de Máquina de Estados Finita e possui múltiplas abas com tabelas e dados de acordo com o WOOM, a fim de oferecer informações detalhadas das ações e interações entre diferentes partes do sistema embarcado ao desenvolvedor do projeto. O artefato desenvolvido oferece as seguintes operações: exportação do diagrama de uma Máquina de Estado Finita, criada no *Modelio* no formato de arquivo SCXML, para que ele possa ser tanto lido e interpretado pela própria extensão quanto possa ser empregado em outras aplicações; uma tabela de “Classes”, que é escrita pelo usuário de acordo com os informações obtidas no passo “Identificar documentos de processo” do método mencionado; uma tabela de “Casos de Uso”, que é populada automaticamente com o conteúdo obtido no arquivo SCXML; e “WARC”, que é uma tabela que reúne todas as informações previamente coletadas e exibe todas as ações e reações do sistema, os responsáveis e os colaboradores dela. A partir dessas informações, é possível escrever e executar testes sobre cada ação e reação esperada, sempre realizando verificações e validações sobre cada aspecto do sistema embarcado a ser projetado. O próximo passo seria refazer a extensão como um módulo de *Modelio*, o que necessitaria de uma conversão do código de Python para Java.