

FERRAMENTA DE AUXÍLIO À AUTOMAÇÃO DE TESTES DE INTERFACES GRÁFICAS DESENVOLVIDAS COM C++

Aluno: Carla Galdino Wanderley
Orientador: Arndt Von Staa

Introdução

A utilização de testes automatizados nos permite verificar com frequência a corretude de sistemas, o que é essencial para garantir a qualidade do software antes de ser liberado para uso. Com o intuito de auxiliar a etapa de testes de sistemas baseados em interface gráfica, foi desenvolvida uma ferramenta que transforma a descrição de um cenário bem definido em um teste de sistema. O teste gerado verifica o comportamento descrito no cenário, aplicando seus passos sobre a interface gráfica.

Objetivos

Disponibilizar uma ferramenta que auxilie na geração e manutenção de testes automatizados para sistemas baseados em interface gráfica de programas redigidos em C++ com apoio da biblioteca QT. A utilização da ferramenta criada deve diminuir o esforço com a criação, execução e manutenção de testes para este tipo de aplicação, já que as suítes de teste são geradas de forma automática. Além disso, visamos o aumento na fidedignidade do software sob teste, através da validação do comportamento implementado definido ao especificar as interfaces com o usuário.

A Ferramenta

Com o objetivo de auxiliar a geração de testes para sistemas desenvolvidos com C++, criamos uma ferramenta que transforma a descrição de um cenário de uso, em um teste da interface gráfica de sistema. Cada cenário deve apresentar um roteiro de passos composto por estímulos e verificações que devem ser realizados sobre a interface do sistema para simular e validar seu comportamento.

Uma das dificuldades apresentadas na geração de testes para interfaces gráficas é a simulação de estímulos para os seus componentes, estímulos estes que normalmente são gerados por dispositivos físicos como teclado e mouse sob controle do usuário humano. Além disso, a forma de testar aplicações através de interfaces gráficas depende fortemente da linguagem de programação utilizada no processo de desenvolvimento. Aplicações que não interagem com um browser são ainda mais difíceis de testar, como é o caso de aplicações desenvolvidas na linguagem C++.

Instruções de Uso

Primeiramente, devemos criar um arquivo de descrição de cenários para servir de entrada para a ferramenta. Em seguida, execute a ferramenta e digite o comando “select file” e logo após este passo, forneça o caminho e nome do arquivo de descrição de cenários a ser utilizado. Se obtiver as seguintes mensagens “XML sucessfully parsed” e “Code generated on /GeneratedTestFiles directory”, o código de teste foi gerado sem erros.

Resultados

A ferramenta gera um arquivo C++ contendo uma classe para cada um dos casos de teste presentes no arquivo XML.

A etapa seguinte à geração dos testes é a incorporação deste arquivo a um projeto que execute testes de unidade. Desta forma, diminuimos o custo de geração e manutenção de testes para o comportamento da interface gráfica do sistema.

Conclusões

A ferramenta tem como objetivo a redução de esforço na criação de suítes de teste durante o processo de desenvolvimento de software. O cliente deve escrever arquivos XML de descrição de cenários que servirão de entrada para a ferramenta. Como a redação de arquivos XML pode ser uma fonte de defeitos, deve ser desenvolvida uma linguagem mais amigável para humanos leigos em computação e que facilite ao usuário redigir os cenários. Este texto será transformado automaticamente para o arquivo XML desejado. Para isso, seria necessário o desenvolvimento de uma ferramenta que transforme cenários escritos em uma linguagem amigável em arquivos XML de descrição de cenários, conforme o formato especificado como entrada para a ferramenta UiUnit. A ferramenta proposta pode ser um trabalho futuro relacionado ao tema.

O arquivo de descrição de cenários utilizado pela ferramenta UiUnit também pode ser obtido através de tabelas de decisão. Para isso, podemos utilizar a ferramenta de apoio à geração de casos de teste desenvolvida por [Caldeira, 2010].

Observamos também que tanto a criação quanto a realização de testes tem um custo baixo, considerando que a criação de um teste para validar uma nova funcionalidade implica apenas a criação de uma coletânea de cenários. Outra vantagem é a facilidade na co-evolução dos testes ao longo do desenvolvimento do software, em que a inserção de novos componentes de interface gráfica não afetará os testes existentes. Este é um ponto que favorece a utilização da ferramenta UiUnit, se comparada com ferramentas de capture & replay, já que a ferramenta desenvolvida não é baseada na representação gráfica da interface, mas em sua estrutura lógica.

Finalmente, este mecanismo pode aumentar o nível de fidedignidade dos sistemas, uma vez que os testes tem o comportamento do sistema – teste de aceitação – como foco principal. Dessa forma, reduzimos a possibilidade de defeitos de sistema persistirem ao final do desenvolvimento.

Referências

1 - CALDEIRA, L. R. **Um experimento em automação de testes com base em casos de uso e com apoio de tabelas de decisão**. Rio de Janeiro, RJ. 2009. Dissertação de Mestrado-Departamento de Informática, Pontifícia Universidade Católica (PUC-Rio).