ENCOSIS
ENCONTRO REGIONAL DE COMPUTAÇÃO E SISTEMAS DE INFORMAÇÃO

2016
ANO V - EDUCAÇÃO TECNOLÓGICA

Minicurso - Automatização de teste de software utilizando a ferramenta SikuliX

Vanilton Pinheiro
vanilton18@gmail.com

FPFtech

Apresentação

- Profissional **Pós-Graduado em Engenharia de Software com Ênfase em Desenvolvimento Web** pela Uninorte em 2015.

- **Bacharel em Ciência da Computação** pela Uninorte em 2013, trabalha como Analista de Teste a mais de 4 anos na FPF Tech. Experiência em VV&T através de Planejamento de teste, especificação, execução e geração de resultados de teste em arquiteturas Web, Móveis e Desktop.

- Atualmente exercendo o papel de Líder de Teste na FPFTech

- Site: http://vanilton.net
- Blog: http://vanilton.net/blog

O que esperar do Curso

Compreender e Aprender Básico da ferrmenta SikuliX IDE

Aprender Básico da SikuliX API Java

Criar scripts de automação com um framework de teste unitário

O que vamos precisar para o Curso

- Java (JDK 1.7+)
- Setup SikuliX IDE
- Dependência para o projeto Java: SikuliX Java API, Junit
- Eclipse IDE (Desenvolvimento projeto em JAVA)

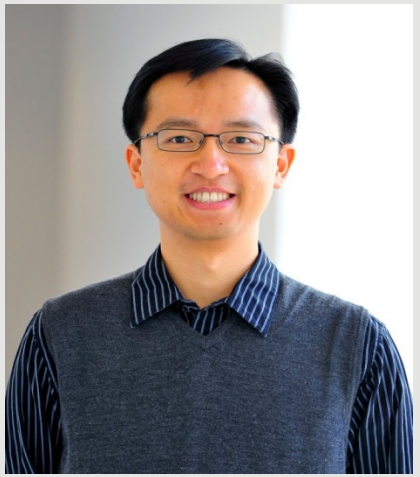
Sikuli

História Sikuli

- O sistema foi projetado sob a **licença MIT** pelo professor **Rob Miller**, estudante de graduação **Tsung-Hsiang Chang** e **Tom Yeh** da Universidade de Maryland.

- **Sikuli** foi iniciado no ano de 2009 como um projeto de pesquisa de código aberto no **Design Interface User Group no MIT**. Tom Yeh e Tsung-Hsiang em 2012 deixaram o projeto Sikuli na versão **Sikuli-X-1.0rc3**.

Rob Miller



Tom Yeh

Tsung-Hsiang Chang

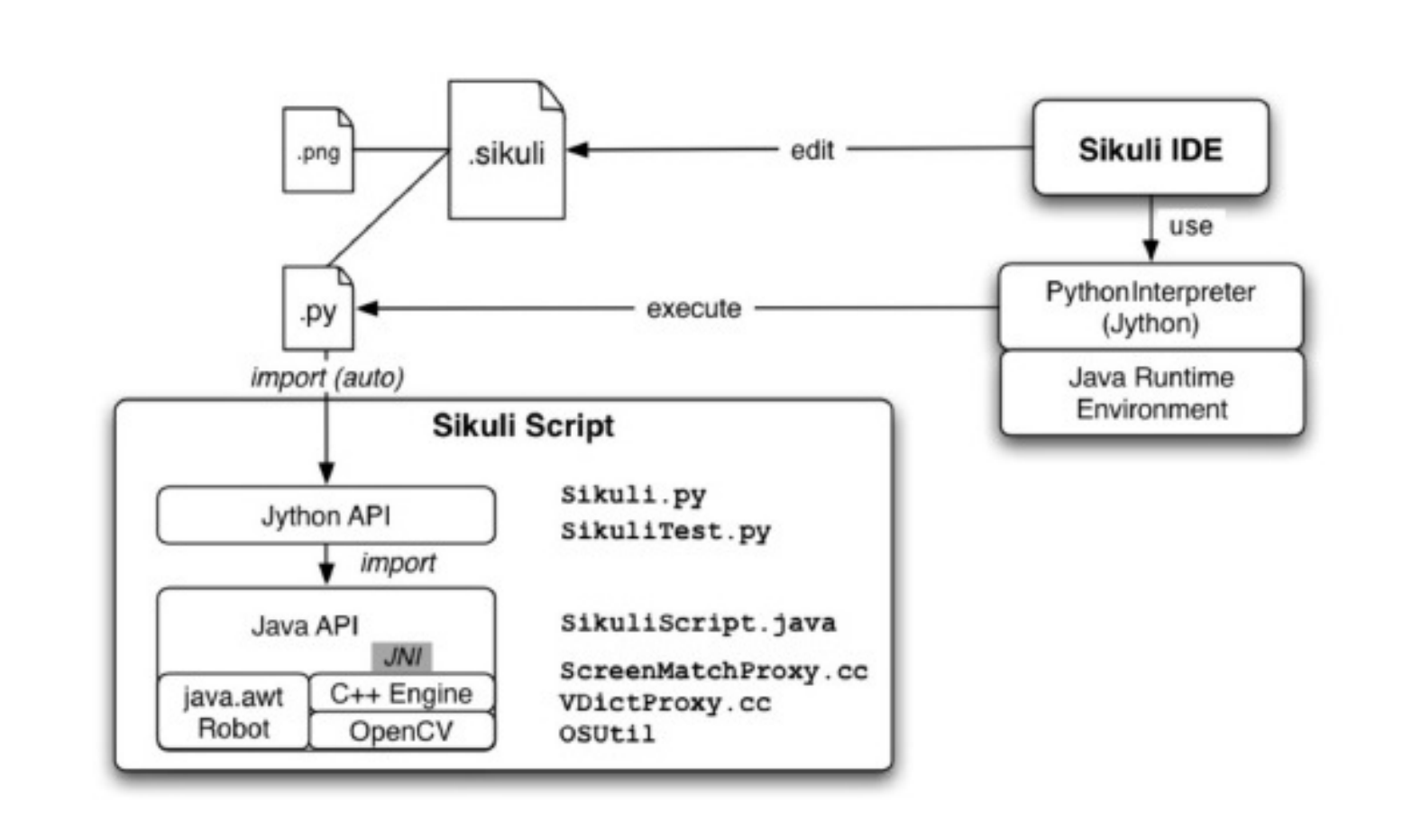
História Sikuli

- O primeiro artigo publicado para a ferramenta foi **Sikuli: Using GUI Screenshots for Search and Automation** *ACM Symposium on User Interface Software and Technology*, October 2009. **(Best Student Paper Award)**
- O Sikuli foi desenvolvido para atuar em um ambiente de script baseado em captura de tela que pode sinalizar um novo paradigma de programação que utiliza a interface gráfica como uma espécie de API.

Curiosidades

- **Sikuli** em Huichol indiano significa olhos de Deus, a capacidade de ver e entender.
- Tom Yeh define o Sikuli com o lema “**O que você vê é o que você codifica**”. Baseado numa metáfora GUI de “**O que você vê é o que obtém**“.

Arquitetura Sikuli



Sikuli atualmente

- **Sikuli** nos dias atuais é mantido e desenvolvido pelo Laboratório Sikuli da Universidade de Colorado Boulder. Ele é apoiado em parte pela Fundação Nacional de Ciência sob o número prêmio IIS-0447800 e pela Quanta Computer, como parte do projeto Tparty.
- Atualmente as atualizações de Sikuli IDE e Script são frequentes atualizadas pelo alemão **Raimund Hocke**, foto ao lado.
- O mesmo está criando um projeto em paralelo chamado **SikuliX**, que engloba tudo referente a Sikuli IDE e Scripts, o mesmo pode ser localizado em http://www.sikulix.com.

Raimund Hocke
RaiMan

Comandos SikuliX IDE – Ações do Mouse

- **click**(imagem)
  - Clica numa dada imagem com o botão esquerdo do mouse.
- **rightClick**(imagem)
  - Clica numa dada imagem com o botão direito do mouse.
- **doubleClick**(imagem)
  - Realiza a ação de dois cliques numa dada imagem com o mouse.
- **hover**(imagem)
  - Cobre uma dada imagem com o ponteiro do mouse.
- **dragDrop**(img1, img2)
  - Arrasta o objeto definido na img1 para a área definida na img2.

Comandos SikuliX IDE – Ações do Teclado

- **type(imagem,texto)**
  - Digita o texto passado na área definida na imagem.
- **paste**(imagem,texto)
  - Cola o texto passado no parâmetro na área definida na imagem.
- **type**(texto)
  - Digita o texto passado onde está o foco do cursor do mouse.
- **paste**(texto)
  - Cola o texto passado no parâmetro onde está o foco do cursor do mouse.
- **type**(Key.ENTER)
  - Realiza a ação da tecla ENTER em um dado local com foco.
- **type("d", Key.META)**
  - Realiza a ação em conjunto das teclas "Windows + d". (Atalho do Windows que minimiza todas as janelas)

Comandos SikuliX IDE – Funções de Localização

- **exists(imagem)**
  - essa função verifica se é possível encontrar a imagem que está como parâmetro. Se for possível, uma ação pode ser realizada, caso contrário, poderá ser executada outra ação de preferência.
- **wait**(imagem, Tempo de espera)
  - essa função recebe como parâmetro uma imagem e, opcionalmente, pode receber também um tempo máximo de espera. É responsável por esperar que a imagem parâmetro seja encontrada, para somente depois continuar com suas ações.
  - waitVanish(imagem, Tempo de espera)
  - tem o objetivo oposto da função wait(Imagem, tempo de espera). Essa função espera a imagem desaparecer para continuar uma ação.
- **find**(imagem)
  - função que permite procurar uma única imagem que pareça com a que está no parâmetro. A Imagem em parâmetro pode ter uma pequena diferença com alguma imagem existente que, mesmo assim, a função a irá encontrar.

Comandos SikuliX IDE – Funções de Localização

- **findAll**(imagem)
  - mesmo objetivo da função find(Imagem), só que procura em vários pontos diferentes por uma imagem igual à que foi passada como parâmetro. Essa função pode considerar uma porcentagem maior de diferença entre a imagem parâmetro e a encontrada para considerá-la como válida.

Comandos SikuliX IDE – Funções observatórias de eventos

- **onAppear**(Imagem, Handler)
  - essa função fica aguardando para ser executada somente quando aparecer uma área igual à área que está representada pela figura em parâmetro. O parâmetro Handler pode ser uma nova função que somente é executada quando a imagem do parâmetro aparecer na tela.
- **observe()**
  - função responsável por observar determinada área a fim de permitir ao criador do script tomar alguma decisão.
- **onChange(**handler)
  - essa função tem por objetivo realizar uma determinada tarefa sempre que uma modificação for realizada em alguma região pré-definida.
- **onVanish**(Imagem, Handler)
  - mesmo objetivo da função onAppear(Imagem, Handler), com a diferença de que esta espera a área correspondente à imagem parâmetro sumir para depois executar uma ação qualquer.

Comandos SikuliX IDE – Outras Funções

- **App.open**(path)
  - essa abri uma aplicação passando como parâmetro seu caminho.
  - **popup**(‘texto’)
  - Essa função quando utilizada retorna um popup na tela, e a continuação das demais instruções fica pausada enquanto não finalizar o popup.
- **print**(‘texto’)
  - essa função quando utilizada exibi no console da Sikuli IDE o texto passado como parâmetro.

Limitações

- Necessidade de ambiente estar **visível na tela**.
- **Focus** nas aplicações a serem testadas.
- **Resoluções de tela** distintas
- **Similaridade**

Prática 1 – Find e Click

1. No app exemplo automatize o processo de acessar o menu funções e abertura da opção Find e Click.
2. Com o passo 1 concluído, automatize o processo de validação dos radio buttons selecionados conforme a imagem do Sistema Operacional.

20 minutos

Prática 2 – Input e Paste

1. No app exemplo automatize o processo de acessar o menu funções e abertura da opção Input e Paste.
2. Com o passo 1 concluído, automatize de validação se o texto digitado utilizando o Input ou Paste, e igual a “vanilton.net”, faça o teste utilizando um valor igual e outro diferente.
3. Após realizar as validaçoes do passo 2, faça o processo de limpar e fechar a tela de "Teste Input e Paste.

20 minutos

Prática 3 – Drag And Drop

1. Acesse a página do Exercício de Drag And Drop

- http://html5demos.com/drag

15 minutos

Referências

- http://www.sikulix.com/
- http://vanilton.net/blog

ENCOSIS
ENCONTRO REGIONAL DE COMPUTAÇÃO E SISTEMAS DE INFORMAÇÃO

2016
ANO V - EDUCAÇÃO TECNOLÓGICA

Minicurso - Automatização de teste de software utilizando a ferramenta SikuliX

Vanilton Pinheiro
vanilton18@gmail.com

FPFtech