

Análise de viabilidade do desenvolvimento de framework na avaliação automática de acessibilidade em RIA

Viability analysis of the development of RIA accessibility evaluation tools

RESUMO

Ao decorrer das últimas décadas, é crescente a utilização de aplicações web para realizar tarefas do dia a dia que antes era realizadas fora desse meio. Com tamanha utilização, cada vez mais interfaces desses sistemas se aprimoram para entregar uma melhor experiência ao usuário. Em decorrência disso, o surgimento de widgets dinâmicos acarretou em grandes dificuldades para usuários que utilizam tecnologias assistivas em utilizar tais aplicações, pelo fato de leitores de telas não saberem interpretar esses widgets dinâmicos. Para entregar um sistema acessível, desenvolvedores deveriam implementar a recomendação ARIA. Entretanto, não é comum desenvolvedores seguirem essa recomendação. Este trabalho propõe analisar a viabilidade tecnológica necessária para desenvolver um controlador de eventos RIA e um robô web, propostos para criação de um framework que auxilie desenvolvedores a implementarem RIA acessíveis.

PALAVRAS-CHAVE: Acessibilidade. ARIA. Widgets.

ABSTRACT

IN THE PAST DECADES, WEB APPLICATIONS USAGE INCREASED AND DAY-TO-DAY TASKS ARE COMMONLY CONDUCTED IN THIS ENVIRONMENT. IN THIS CONTEXT, EVEN MORE COMPANIES ARE CONCERNED WITH DELIVERING A BETTER EXPERIENCE TO USERS. DUE TO THAT, THE EMERGENCE OF DYNAMIC WIDGETS CAUSES MANY DIFFICULTIES TO USERS THAT USE ASSISTIVE TECHNOLOGY TO ACCESS WEB APPLICATIONS, GIVEN THAT SCREEN READERS DON'T KNOW HOW TO INTERPRET THESE DYNAMIC COMPONENTS. TO DELIVER AN ACCESSIBLE SYSTEM, DEVELOPERS MUST IMPLEMENT THE ARIA RECOMMENDATION. HOWEVER, DEVELOPERS ARE NOT ALWAYS AWARE OF THESE SPECIFICATIONS. THIS WORKS AIMS TO ANALYSE THE TECHNOLOGICAL VIABILITY OF DEVELOPING ARIA EVENTS CONTROLLER AND A WEB ROBOT, WHICH ARE NECESSARY COMPONENTS FOR THE DEVELOPMENT OF A FRAMEWORK THAT ASSIST DEVELOPERS TO IMPLEMENT ACCESSIBLE RIA.

KEYWORDS: Accessibility. ARIA. Widgets.

Gabriel Romero de Souza
gabrielromerods@hotmail.com
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil
Willian Massami Watanabe
wwatanabe@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

A crescente utilização da Internet para realização de diversas tarefas como pagar contas, realizar compras, entre outras acarretou na necessidade de melhorar a apresentação das aplicações *web*. Páginas *web* deixaram de apresentar seu conteúdo de forma estática e evoluíram para *layouts* dinâmicos respondendo de acordo com a navegação do usuário para entregar uma melhor experiência. Usuários sem necessidades especiais conseguem usufruir dessas melhorias. Entretanto, a complexidade dessa apresentação dinâmica de conteúdo acarretou em problemas para tecnologias assistivas em interpretar e informar aos usuários deficientes sobre os conteúdos dinâmicos (GIBSON, 2007). Por consequência, leitores de telas utilizados por usuários deficientes não reconhecem esses conteúdos dinâmicos, os *widgets* dinâmicos, podendo passar despercebidos por esses usuários ou impossibilitando a utilização da aplicação.

Para contornar esse dilema, a *Web Accessibility Initiative* (WAI) elaborou a especificação *Accessibility Rich Internet Applications* (ARIA) (World Wide Web Consortium, 2019) para auxiliar leitores de telas a reconhecer tais *widgets* dinâmicos assim podendo comunicar ao usuário informações relevantes do elemento como qual tipo de *widget* ele está manipulando, quais as funcionalidades disponíveis e como navegar pelo *widget*. No entanto, não há nenhuma garantia de que os desenvolvedores *web* sigam a especificação.

Este trabalho teve como objetivo investigar a viabilidade tecnológica da implementação do *framework* proposto por (DOUSH et al., 2013) com foco no desenvolvimento de alguns componentes presentes na proposta. O propósito do *framework* é disponibilizar uma ferramenta que avalie automaticamente acessibilidade em sites, gerando um relatório para que desenvolvedores possam avaliar falhas de acessibilidade presentes.

MATERIAL E MÉTODOS

Para a realização do trabalho, foram utilizados a biblioteca de testes automatizados Selenium e a *API Mutation Observer* com a linguagem Java. Através dessas tecnologias, foi possível o desenvolvimento do componente robô *web*.

A biblioteca Selenium possibilita a automatização de navegadores *web* e simular ações que um usuário pode efetuar na página como exemplo click do botão direito do mouse, click e segurar, posicionar o cursor do mouse em cima de um elemento, etc. Ao simular ações do usuário, mutações na estrutura da página podem ocorrer. A *API mutation observer* possibilita monitorar mudanças feitas no *Document Object Model* (DOM) responsável pela estrutura da página, retornando os elementos

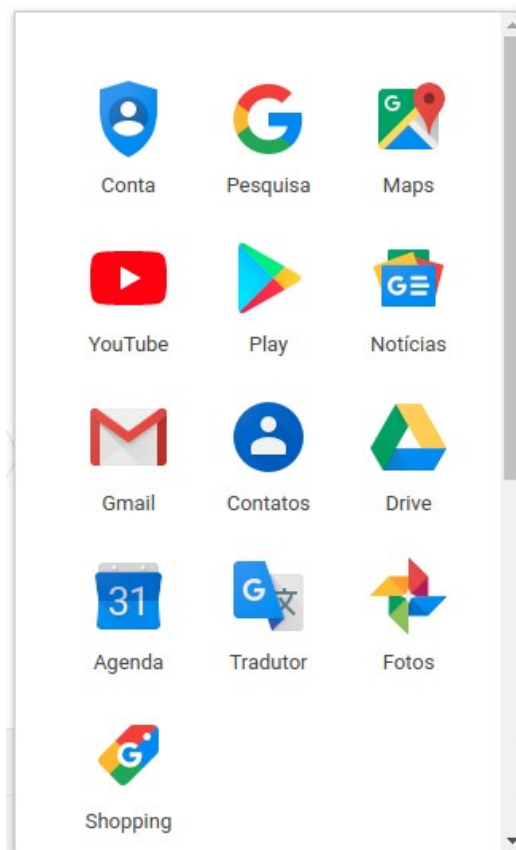
mutantes gerados pela ação. Com o retorno das mutações, é possível capturar informações a respeito delas para avaliação.

O componente robô *web* realiza ações simulando o comportamento do usuário na página, passando por todos os elementos da página. Ao passar pelo elemento, é efetuado duas simulações de ações, uma de clicar no elemento com botão esquerdo do mouse e a outra de posicionar o cursor no mouse sobre o elemento. Além da simulação, o robô *web* ao capturar uma nova mutação registra uma *screenshot* da mesma e salva informações a respeito da mutação.

RESULTADOS E DISCUSSÃO

O robô *web* ao passar por algum elemento que altera a estrutura da página, captura uma *screenshot* da(s) mutação(ões) ocorridas independente de qual tipo de ação que ativou a mutação, como exemplo a Figura 1.

Figura 1 - Mutação da tela inicial do Google



Fonte: Google (2019).

Com as imagens das mutações, é possível analisar quais tipos de *widgets* foram capturados. Além das *screenshots*, com a API *mutation observer* informações sobre as mutações são registradas. Na Figura 2 é possível analisar os dados capturados em um arquivo do formato JSON, onde além das informações das

mutações, é registrado informações do elemento responsável pelas mutações.

Figura 2 - Dados capturados

```
{
  "Height": 30,
  "Ação": "Mouse ouver",
  "Info mutation": [
    {
      "InnethHTML": 651,
      "Left": 942,
      "Mutacao": 0,
      "Tag": "DIV",
      "Top": 56,
      "XPath": "\/body[2]\/div[4]\/div[4]\/div[1]\/div[1]\/div[1]\/div[3]"
    },
    {
      "InnethHTML": 0,
      "Left": 1129,
      "Mutacao": 0,
      "Tag": "A",
      "Top": 16,
      "XPath": "\/body[2]\/div[4]\/div[4]\/div[1]\/div[1]\/div[1]\/div[2]\/div[1]\/div[1]\/a[1]"
    }
  ],
  "Elemento": 6,
  "Localização x": 985,
  "Width": 313,
  "Localização y": 16
},
```

Fonte: Autoria própria.

Na propriedade *info mutation* são registrado os dados das mutações. Os dados capturados das mutações representam a posição em que se encontra a mutação, com o dados *Left* e *Top*, o *Xpath* do elemento, qual tipo de elemento HTML no campo *Tag* e o campo *InnerHTML* registra o número de caracteres que elemento possui. Além das informações das mutações, as informações fora da propriedade *info mutation* diz respeito ao elemento que ativou as mutações. Analisando a Figura 2, é possível saber qual tipo de ação ativou as mutações, a localização e tamanho do elemento acionador.

As informações e *screenshots* registradas pelo robô *web* podem ser utilizadas por outros componentes do *framework*. Elas auxiliam a classificar qual tipo de *widget* é a mutação para assim informar leitores de telas em como lidar com o elemento.

CONCLUSÃO

A adesão de aplicações web para fornecer serviços, faz com que seja importante fornecer sistemas acessíveis a todos. A especificação ARIA possibilita que leitores de telas consigam lidar com *widgets* dinâmicos, assim possibilitando repassar ao usuário informações sobre o *widget*. No entanto, não existe nenhuma garantia que desenvolvedores *web* sigam as recomendações ARIA. O propósito do *framework* proposto por (DOUSH et al., 2013) é auxiliar desenvolvedores a implementarem tais recomendações.

O desenvolvimento do componente robô *web* presente neste trabalho agrega ao desenvolvimento do *framework*. Com o

componente desenvolvido, é possível capturar elementos mutantes decorridos de uma ação na aplicação e informações desse elemento, que poderam ser utilizadas por outros componentes presentes no *framework*.

AGRADECIMENTOS

O presente trabalho foi realizado com o apoio da Fundação Araucária FA - Paraná/Brasil.

REFERÊNCIAS

GIBSON, B. Enabling an accessible web 2.0. In: **Proceedings of the 2007 International Cross-disciplinary Conference on Web Accessibility (W4A)**. New York, NY, USA: ACM, 2007. (W4A 07), p. 1-6. ISBN 1-59593-590-8. Disponível em: <https://doi.acm.org/10.1145/1243442>.

DOUSH, I. et al. The design of ria accessibility evaluation tool. **Advances in Engineering Software**, v. 57, p. 1-7, 03 2013.

World Wide Web Consortium. **Accessible Rich Internet Application (WAI-ARIA) 1.1**. Disponível em : <https://www.w3.org/TR/wai-aria-1.1/>. Acesso: 17 ago. 2019.

Google. **Google**. Disponível em: <https://www.google.com>. Acesso: 17 ago. 2019.