

**UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
RÉGIS DINIZ CARREIRO**

**Análise e otimização da avaliação de acessibilidade em
ambientes remotos através de confronto de comportamentos**
(ainda não tenho certeza se vou deixar esse título)

**São Paulo,
2010**

**UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
RÉGIS DINIZ CARREIRO**

**Análise e otimização da avaliação de acessibilidade em
ambientes remotos através de confronto de comportamentos**
(ainda não tenho certeza se vou deixar esse título)

*Monografia apresentada ao Instituto de Matemática e Estatística
para conclusão do Bacharelado em Ciência da Computação*

Área de concentração:
Acessibilidade Digital

Orientadores:
*Profa. Dra. Lúcia Vilela Leite Filgueiras
(Escola Politécnica, USP)
Prof. Dr. Marcelo Morandini
(Escola de Comunicações e Artes, USP)*

**São Paulo,
2010**

AGRADECIMENTOS

Alguma epígrafe aqui...

RESUMO

Quando concluída a monografia, escrever aqui o resumo.

SUMÁRIO

PARTE SUBJETIVA

1. INTRODUÇÃO

- 1.1 Acessibilidade e deficiências**
- 1.2 Objetivo**
- 1.3 Motivação**
- 1.4 Metodologia**
- 1.5 Estrutura**

2. ACESSIBILIDADE NA WEB

- 2.1 Leitores de Tela**
- 2.2 Avaliações de Acessibilidade**
 - 2.2.1 Avaliação remota automática de código**
 - 2.2.2 Avaliação do uso através de gravação**
 - 2.2.3 Os problemas desses métodos**

3. METODOLOGIA

- 3.1 Estudo do problema de acessibilidade e sua avaliação**

4. O SOFTWARE

- 4.1 Etapas da construção**
- 4.2 Funcionamento**
- 4.3 Provando resultados: O ErgoCoin**
- 4.4 Aprimoramentos futuros**

5. CONCLUSÃO E DISCUSSÃO DE RESULTADOS

6. BIBLIOGRAFIA

PARTE SUBJETIVA

PARTE OBJETIVA

1. INTRODUÇÃO

1.1 Acessibilidade e deficiências

Acessibilidade é um termo genérico usado para descrever o grau em que um produto, dispositivo ou ambiente é acessível ao máximo número possível de pessoas. A acessibilidade pode ser vista como a “capacidade de acessar” de um indivíduo e deste, a partir daí, se beneficiar de algum sistema ou entidade. A acessibilidade é frequentemente focada em pessoas com deficiências e o direito destas ao acesso às entidades, este usualmente feito por meio de tecnologias assistivas. (<http://en.wikipedia.org/wiki/Accessibility>).

Existem diversos tipos de deficiências, listadas na tabela abaixo, e para cada tipo uma série de tecnologias assistivas que buscam tornar possível a pessoa portadora o acesso de que esta necessita.

[fazer uma tabela comparativa descritiva das deficiências para contextualizar o leitor]

Eyesight Disability
Hearing Disability
Mobility Disability
Cognitive Disability
Mental
Múltipla
Aprendizado

Cada um destes tipos de deficiência se subdivide em grupos menores e para cada um destes grupos existe uma série de tecnologias assistivas. Devido a uma questão de escopo, nos focaremos no presente trabalho nas tecnologias assistivas para acesso web por deficientes visuais totais.

1.2 Objetivo

O objetivo do presente trabalho é fazer um estudo comparativo das ferramentas existentes para a avaliação de acessibilidade na web e propor uma metodologia alternativa - o WebModelEval (nome do soft?) - , bem como comparar o resultado dessa avaliação com o resultado das outras ferramentas.

1.3 Motivação

A motivação para esse trabalho surgiu após algumas experiências que tive trabalhando com acessibilidade na web na Universidade de São Paulo. Inicialmente um completo leigo no assunto, fui convidado pela coordenadora da rede Saci (link – definir onde ficarão os links) Ana Maria Barbosa e pela bacharel em Ciência da Computação pelo IME-USP, Lucy Gruenwald, a desenvolver,

junto com uma equipe de programadores, um site de vagas e currículos (link) que fosse completamente acessível e tivesse como foco o usuário com deficiência.

Não foi necessário muito tempo para que me fosse possível perceber o quão inaptos estão a maioria dos desenvolvedores web do Brasil e quão pouco cuidado é tomado hoje por estes para que se construam plataformas online verdadeiramente acessíveis para as pessoas com deficiência. A associação da falta de profissionais comprometidos com a causa no Brasil bem como meu interesse em aprender mais sobre o assunto culminaram em minha participação no Accessing Higher Ground 2009 (explicar melhor o que é a conferência?), em Westminster, Colorado, EUA.

No começo do ano fui convidado novamente pela Lucy Gruenwald a participar de mais um projeto que visava oferecer recomendações de acessibilidade ao PECE (explicar o que é o PECE, Poli). Neste contexto conheci a professora Lucia Filgueiras – que posteriormente apresentou-me ao professor Marcelo Morandini – e que decidiu orientar-me neste trabalho. Alguns resultados deste último trabalho ajudam a suportar conceitos neste TCC.

[falar que a ferramenta é inovadora aqui ou mais adiante?]

1.4 Metodologia para definição do trabalho

O método de trabalho basicamente consistiu em pesquisa das alternativas existentes e desenvolvimento do software a partir da definição de sua viabilidade.

[melhorar muito isso aqui...]

1.5 Estrutura

2. ACESSIBILIDADE NA WEB

Aprofundando um pouco o conceito de acessibilidade visto no item 1.1, agora com foco na web, podemos inferir que uma web acessível é aquela que permite acesso completo às suas informações e serviços a todas as pessoas, sejam elas portadoras de algum tipo de deficiência ou não.

Não é preciso muita reflexão para que se entenda a importância de se ter uma web acessível, visto que cada vez mais a rede está sendo usada para compras, educação, leitura de notícias etc. Construir plataformas que não contemplem as pessoas com deficiência é certamente uma forma de exclusão que vai se tornando cada vez mais grave a medida que os anos vão passando e o uso da internet vai se tornando mais e mais relevante em nosso cotidiano.

Como o escopo deste trabalho é das pessoas com deficiência visual total [é? dizer isso melhor antes de chegar aqui...], precisamos entender de que tipo de tecnologia assistiva esse usuário faz uso e o que estrutura representa para ele uma site acessível.

2.1 Leitores de Tela

Os leitores de tela são a ferramenta pela qual os deficientes visuais totais conseguem utilizar o computador. Basicamente o programa lê as instruções da janela ativa e fala para usuário para que este consiga entender onde está, o que está fazendo e quais informações estão na tela no momento.

Os mais conhecidos no mercado são:

- Dosvox
- Jaws
- NVDA
- Virtual Vision
(disponibilizar links)

Na navegação pela internet esses softwares auxiliam lendo toda a página para o usuário, através da interpretação correta da marcação html e provêem uma quantidade razoável de recursos para que o internauta possa confortavelmente navegar pelo site.

A maioria desses softwares hoje já se sofisticaram o suficiente para conseguir oferecer recursos como leitura de links e navegação por headers.

2.2 Avaliações de Acessibilidade

Como foi dito anteriormente, um leitor de tela na web interpreta o código html e fala de maneira apropriada, a partir da interpretação desse código, as instruções para a navegabilidade do usuário. Como é de se esperar, se esse código não for escrito corretamente, por melhor que seja software, a página vai se tornar completamente inacessível. Um exemplo claro desse problema é uma imagem que se apresenta em uma página sem um texto alternativo. Se essa imagem for crucial para o entendimento do conteúdo em questão e não for provido para a mesma uma boa descrição, é bastante provável que acessibilidade do conteúdo para deficientes visuais fique consideravelmente comprometida.

Mas como construir uma página acessível a deficientes visuais e como saber se uma página já construída é acessível?

O W3C (World Wide Web Consortium) divulgou em [checar data] a WCAG 2.0 (Web Content Accessibility Guidelines) para que os desenvolvedores pudessem entender melhor como construir uma página acessível às pessoas com deficiência. A análise desse documento item a item, com foco na deficiência visual total, será feita em momento posterior nesse texto.

Estando concluída a página, como saber, entretanto, se a mesma é acessível?

Basicamente a avaliação de acessibilidade de uma página consiste na análise da mesma para verificação de que a possibilidade de uso pleno de suas funcionalidades e extração de toda informação ali existente é possível a pessoa com deficiência.

Atualmente existem dois métodos bastante conhecidos para essa avaliação.

2.2.1 Avaliação remota automática de código

Consiste em uma ferramenta que avalia o código html da aplicação checando se nenhuma das normas da WCAG são quebradas. Por exemplo, verifica se todas as imagens da página tem um texto alternativo etc.

As ferramentas mais conhecidas para essa avaliação são

WAVE ([link](#))

Cynthia Says ([link](#))

DaSilva ([link](#))

Para que o leitor tenha noção do que é de fato o teste basta entrar em uma das urls acima e solicitar a avaliação de algum site que achar conveniente.

2.2.2 Avaliação do uso através de gravação

Consiste basicamente da gravação da navegação do usuário e de seu posterior envio para análise por um especialista em acessibilidade. É muito efetivo, mas muito custoso.

Realizamos alguns testes desse tipo no portal do PECE. Para que o leitor entenda de fato que tipo de teste é esse basta acessar: (Já pedi uma cópia dos vídeos dos testes que fizemos para o Marcelo – que tem acesso ao servidor – mas ele ainda está providenciando. Preciso ver se temos autorização para mostrar os vídeos for a da Poli) e assistir alguns dos vídeos de testes.

2.2.3 Os problemas desses métodos

Infelizmente os dois métodos citados apresentam problemas. A avaliação automática de código é remota e barata, entretanto, por ater-se apenas a avaliação de código peca por não poder avaliar elementos que estão além desse conceito. Por exemplo, se uma imagem possui um texto alternativo mas este, entretanto, não define de fato a imagem eficientemente, o avaliador apontará que está tudo correto, pois considera apenas válida a existência do texto alternativo, não tendo como avaliar a efetividade do mesmo.

O método de gravação é muito eficaz, entretanto, muito custoso. Imagine só a complexidade e tempo gastos para avaliar um a um os testes dos usuários (que levam vários minutos) e diagnosticar os problemas relacionados a todas as suas idiossincrasias.

3. METODOLOGIA

O objetivo deste trabalho é ao menos ensaiar o desenvolvimento de uma ferramenta que sirva como alternativa mais eficaz que a avaliação de código e menos cara que a avaliação de gravações (teste de usabilidade- definir usabilidade no começo!!!).

O modelo do software completo, sobre o qual falaremos mais na seção “Desenvolvimento Futuro” seria:

1. Um servidor em que os desenvolvedores de websites pudessem criar uma conta com usuário e senha;
2. Neste mesmo servidor os usuários, ao solicitarem a gravação de comportamentos padrão para ações em seus websites, recebem um código que os permite gravar as ações modelo. Por exemplo, um desenvolvedor coloca o código fornecido em sua página (a exemplo do que é feito com o Google Analytics) e pode gravar uma ação, por exemplo a matrícula em um curso, que será monitorada pelo servidor.
3. O servidor detecta, através de script, sempre que a ação é iniciada e verifica se o comportamento do usuário corresponde ao comportamento gravado pelo desenvolvedor. O servidor compara ainda o tempo para execução da ação e a diferença entre usuários que usam predominantemente o mouse – usuário comum – e aqueles que utilizam exclusivamente o teclado do momento que entram até quando saem do site – comportamento aqui associado ao usuário com deficiência visual.
4. A partir disso o servidor gera relatórios da usabilidade (preciso definir usabilidade no início) e acessibilidade da ação monitorada e oferece ao desenvolvedor para que este possa, se necessário, fazer alterações em seu site.

A preocupação do trabalho foi desenvolver um software que conseguisse gravar as ações dos usuários e compará-las com o modelo gravado pelo desenvolvedor. As questões do servidor, contas de usuário e código para gravação, mais associadas ao modelo de negócio do que a tarefa em si, foram deixadas para um desenvolvimento posterior, na continuidade do projeto.

* Preciso dar um nome para o software ainda...

3.1 Estudo do problema de acessibilidade e sua avaliação

A questão que ficou para responder agora, antes de proceder a o desenvolvimento do software é a seguinte, supondo que tenhamos o software descrito funcionando perfeitamente, a que questões de acessibilidade poderíamos responder com vantagem em relação ao uma avaliador de código comum, que não considera o comportamento de usuário? Em relação à avaliação de gravações talvez a diferença maior não esteja no resultado em si,

mas sim no custo, praticidade e possibilidade de levar em conta a experiência de milhares ou mesmo milhões de usuários, algo praticamente impossível de alcançar com a avaliação de cada gravação por especialistas.

Para responder essa questão utilizamos, como comparativo a nosso software aqui considerado já construído, o avaliador automático de código WAVE. Da renomadíssima instituição que suporta a acessibilidade na web, WebAim. Fizemos então um estudo, a partir de cada ponto da guideline, do que seríamos capazes de responder melhor como nosso software do que com o Wave. (explicar melhor o que é o Wave, prover links)

[É preciso distrinchar mais as Guidelines. Incluir avaliação de subitens? Revisar essa seção. Traduzir o texto?]

Guideline 1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

A ferramenta de avaliação automática consegue verificar no código se um texto alternativo está presente quando se trata, por exemplo, de um elemento gráfico. Entretanto não é possível para a ferramenta dizer o quão bom está este texto para um usuário com deficiência. Por exemplo, se é apresentada uma imagem com instruções de como usar um serviço e o seu texto alternativo for “instruções” a página “passará” pelo avaliador automático. Entretanto, se percebermos que, entre os usuários que utilizam mais o mouse, se costuma observar a imagem (inércia) por cerca de 10s e depois disso executar um determinado número de passos e notarmos que um usuário que navega pelo teclado já começa, em tempo bem menor, a pressionar muitos “tabs” e “enters”, temos aqui um indicativo claro de um problema exclusivo de acessibilidade.

Exemplo bastante prático: captcha.

[Muita vantagem em relação ao WAVE]

Guideline 1.2 Time-based Media: Provide alternatives for time-based media.

O Wave não consegue dizer se um vídeo ou áudio foram de fato acessíveis. Exemplo para vídeo: No caso do nosso sistema, a maneira mais fácil que vejo seria detectar um comportamento de navegação exclusiva via teclado e comparar o tempo de permanência na página do vídeo com o tempo médio de permanência na mesma página por usuários que navegam majoritariamente utilizando o mouse. Uma diferença muito considerável poderia ser um indicativo forte de inacessibilidade do material.

[Muita vantagem em relação ao WAVE]

Guideline 1.3 Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

Avaliação do layout está fora do escopo das possibilidades com avaliadores. [Nem WAVE nem nosso sistema]

Guideline 1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background.

Melhor avaliado com comparação de cor de texto e cor de background, verificando o grau de proximidade de tons.

[Vantagem do WAVE]

Guideline 2.1 Keyboard Accessible: Make all functionality available from a keyboard.

O Wave pode avisar sobre a presença de um elemento de javascript na página. Mas é completamente possível que se use javascript e a página seja plenamente acessível. O Wave, na maioria dos casos, não sabe dizer quando a presença de javascript na página comprometerá ou não a acessibilidade da mesma. Aqui poderíamos avaliar por exemplo se os usuários que navegam por clique obtêm um resultado diferente daqueles que navegam pelo teclado para disparar uma ação. Por exemplo, se existe um link cuja ação está especificada por “onclick”, usuários que sempre usam o mouse tenderão a chegar na página alvo enquanto os que usam apenas o teclado possivelmente não conseguirão nem focar o link.

[Muita vantagem em relação ao WAVE]

Guideline 2.2 Enough Time: Provide users enough time to read and use content.

Aqui existe a possibilidade de verificar, caso na página ocorra algum evento disparado pelo servidor após determinado tempo que mude os elementos do site sem ação do usuário, se após disso o usuário que navega por tabs costuma apertar refresh ou voltar ou se sente perdido (grande inércia ou vários tabs em ciclo), enquanto os usuários que usam mais o mouse não.

[Muita vantagem em relação ao WAVE]

Guideline 2.3 Seizures: Do not design content in a way that is known to cause seizures.

Avaliação dessa característica está fora do escopo de avaliação automática.

[Nem WAVE nem nosso sistema]

Guideline 2.4 Navigable: Provide ways to help users navigate, find content, and determine where they are.

Se a maioria dos usuários com mouse ao estar em uma página costumam executar qualquer ação dentro de determinado intervalo de tempo e em caso de navegação por tabs observamos foco em ciclos por um tempo muito superior à referida média ou refresh constante da página, provavelmente estamos em um caso de navegação inacessível ou com acessibilidade muito ruim.

[Vantagem em relação ao WAVE]

Guideline 3.1 Readable: Make text content readable and understandable

Basicamente essa “guideline” se refere a um texto que seja escrito de maneira inteligível pela maioria dos usuários. Acredito ainda ser muito difícil avaliar este aspecto, relacionado, de certa forma, à web semântica.

[Nem WAVE nem nosso sistema]

Guideline 3.2 Predictable: Make Web pages appear and operate in predictable ways.

Se quando determinada ação é executada é notável a mudança considerável de elementos da página (drástica mudança de layout) e após isso temos o mesmo comportamento definido em 2.4, é muito provável que tenhamos um caso de página operando de modo “imprevisível”.

[Muita vantagem em relação ao WAVE]

Guideline 3.3 Input Assistance: Help users avoid and correct mistakes.

Se é perceptível em uma navegação por tabs uma taxa de retorno a uma página de entrada de dados muito superior à média da navegação dos usuários “clicantes”, provavelmente o sistema não está informando os erros do formulário ao usuário de maneira efetiva ou mesmo o formulário está completamente inacessível.

[Muita vantagem em relação ao WAVE]

Guideline 4.1 Compatible: Maximize compatibility with current and future user agents, including assistive technologies.

A avaliação só é possível por verificação de código.

[Vantagem do WAVE]

Estatísticas

Guidelines: 12

Quem avalia melhor?

Nosso sistema: 8 (66.7%)

WAVE: 2 (16.7%)

Nenhum: 2 (16.7%)

Ou seja, daqui foi possível concluir que a construção do software era de fato válida.

4. O SOFTWARE

4.1 Etapas da construção

4.2 Funcionamento

Explicar jQuery + caminhos em grafos e peso dos caminhos (no futuro) para correção do modelo proposto pelo desenvolvedor, se for o caso.

4.3 Provando resultados: O ErgoCoin

Falar do auxílio inicial do software do Morandini para provar os resultados empíricos.

4.4 Aprimoramentos futuros

Falar da continuidade do projeto e do que eu gostaria que o software se tornasse.

5. CONCLUSÃO E DISCUSSÃO DE RESULTADOS

6. BIBLIOGRAFIA

PARTE SUBJETIVA