UNICAMP

Vagner Figueredo de Santana

“Interfaces Autoajustáveis em *Websites*: Contribuições em Direção ao *Design* para Todos”

“*Self Tailorable Website Interfaces: Contributions Towards the Design for All*”

CAMPINAS
2012

State University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Vagner Figueredo de Santana

“Interfaces Autoajustáveis em *Websites*: Contribuições em Direção ao *Design* para Todos”

Supervisor:
Orientador(a): **Maria Cecília Calani Baranauskas**

“Self Tailorable Website Interfaces: Contributions Towards the Design for All”

PhD Thesis presented to the Post Graduate Program of the Institute of Computing of the State University of Campinas to obtain a PhD degree in Computer Science.

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Doutor em Ciência da Computação.

This volume corresponds to the final version of the Thesis defended by Vagner Figueredo de Santana, under the supervision of Maria Cecília Calani Baranauskas.

Este exemplar corresponde à versão final da Tese defendida por Vagner Figueredo de Santana, sob orientação de Maria Cecília Calani Baranauskas.

Supervisor’s signature / *Assinatura do Orientador(a)*

CAMPINAS
2012

FICHA CATALOGRÁFICA ELABORADA POR
MARIA FABIANA BEZERRA MULLER - CRB8/6162
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA - UNICAMP

Santana, Vagner Figueredo de, 1982-
Sa59i Interfaces autoajustáveis em websites: contribuições em direção ao Design para Todos / Vagner Figueredo de Santana. – Campinas, SP : [s.n.], 2012.

Orientador: Maria Cecília Calani Baranauskas.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Interfaces de usuário (Sistema de computador) - Avaliação. 2. Interação homem-máquina. 3. Acessibilidade. 4. Usabilidade. 5. Internet. I. Baranauskas, Maria Cecília Calani,1954-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em inglês: Self tailorable website interfaces: contributions towards the Design for All
Palavras-chave em inglês:
User interfaces (Computer systems) - Evaluation
Human-computer interaction
Accessibility
Usability
Internet
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Maria Cecília Calani Baranauskas [Orientador]
Fabio Paternò
Renata Pontin de Mattos Fortes
André Santanchè
Eliane Martins
Data de defesa: 28-09-2012
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Tese Defendida e Aprovada em 28 de setembro de 2012, pela Banca examinadora composta pelos Professores Doutores:

E Martins

Profª. Drª. Eliane Martins
IC / UNICAMP

André Santanchè

Prof. Dr. André Santanchè
IC / UNICAMP

Fabio Pater

Prof. Dr. Fábio Paternò
CNR

Prof. Dr. Renata Pontin de Mattos Fortes
ICMC / USP

Profª. Drª. Maria Cecília Calani Baranauskas

Institute of Computing /*Instituto de Computação*
State University of Campinas /*Universidade Estadual de Campinas*

Interfaces Autoajustáveis em *Websites*: Contribuições em Direção ao *Design* para Todos

Vagner Figueredo de Santana[1]

September 28, 2012

Examiner Board/*Banca Examinadora*:

- Maria Cecília Calani Baranauskas (Orientadora)
  Instituto de Computação, Universidade Estadual de Campinas (UNICAMP)
- Prof^{a}. Dr^{a}. Eliane Martins
  Instituto de Computação, Universidade Estadual de Campinas (UNICAMP)
- Prof. Dr. Fabio Paternò
  Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo",
  Consiglio Nazionale delle Ricerche (CNR)
- Prof^{a}. Dr^{a}. Renata Pontin de Mattos Fortes
  Instituto de Ciências Matemáticas e de Computação,
  Universidade de São Paulo (USP)
- Prof. Dr. André Santanchè
  Instituto de Computação, Universidade Estadual de Campinas (UNICAMP)

[1] Suporte financeiro: Bolsa de doutorado FAPESP (processo 2009/10186-9) 2009–2012.

Resumo

Devido ao rápido crescimento da Web nas últimas décadas e à interatividade proporcionada pela Web 2.0, serviços de governo, entretenimento e educação são, cada vez mais, disponibilizados na Internet. No entanto, parte significativa dos esforços que seguem esse crescimento não considera as diferenças existentes em toda a população de usuários.

Existem abordagens e técnicas voltadas para o *design* inclusivo nas fases de análise, projeto, desenvolvimento e avaliação de *websites*. Nas etapas iniciais de análise é possível identificar diversas tarefas e formas de utilização de interfaces de usuário (IUs), mas questões relativas às diferentes necessidades dos usuários podem surgir em diversos contextos de uso não previstos em etapas anteriores à sua utilização real.

Interfaces ajustáveis representam uma forma promissora de promover a acessibilidade e usabilidade na Web e assim possibilitar que necessidades surgidas nos mais diversos contextos de uso sejam consideradas não apenas na análise, no projeto e na implementação, mas também durante a utilização de IUs. Nessa direção, a pergunta de pesquisa que norteou o estudo proposto nesta tese foi a seguinte: é possível desenvolver uma avaliação contínua, baseada em dados detalhados de uso, que gere ajustes à IU tendo como objetivo o *Design* para Todos?

Esta tese visou apresentar abordagens existentes de avaliação de *websites* e propor uma solução que utiliza *logs* de eventos disparados no lado do cliente para identificar comportamentos dos usuários e ajustar a IU automaticamente de acordo com as formas de interação utilizadas pelos participantes. Dado que tal contexto levanta questões envolvendo tanto a plataforma técnica quanto a plataforma social, relacionadas ao desenvolvimento de sistemas de informação, o principal referencial teórico-metotológico seguido é o da Semiótica Organizacional.

O objetivo principal foi possibilitar uma avaliação continuada de *websites*, buscando eliminar barreiras de acessibilidade e problemas de usabilidade. Os resultados obtidos na tese envolvem:

1. formalização da heurística, base para a identificação de incidentes de uso e análise dos ajustes, que mostrou cerca de 65% de efetividade;

2. definição de um conjunto de 22 requisitos para ferramentas de avaliação tendo como base em estudo envolvendo 11 usuários de uma Rede Social Inclusiva (RSI);

3. *design* e *redesign* do relatório apresentado aos avaliadores, contando com a participação de 28 especialistas em tecnologia da informação;

4. proposta de uma taxonomia para classificar ferramentas de avaliação considerando 4 dimensões (i.e., interação entre participante e avaliador, fonte de dados, nível de esforço por parte dos avaliadores e participantes e tipo de automatização fornecida pela ferramenta);

5. identificação de diferenças estatisticamente significativas envolvendo comprimento ($p = \mathit{0.01}$) e presença de mouse ($p = \mathit{0.01}$) nas cadeias de eventos relativas às sessões de pessoas que usam ou não usam tecnologia assistiva; complementarmente, esses dados possibilitaram a classificação dessas cadeias de eventos com mais de 80% de sucesso;

6. levantamento de padrões relativos a como usuários de tecnologia assistiva interagem com *websites* em contexto de uso real;

7. apresentação da viabilidade de se aplicar uma abordagem baseada em normas para gerar e aplicar ajustes automaticamente, considerando dados de uso coletados em contexto real de utilização.

Com resultados desta tese espera-se apoiar abordagens que considerem ajuste automático de *websites* a partir da análise de *logs* de eventos, em conjunto com técnicas de *Web Usage Mining*, considerando diferenças e contribuindo para que a Web se torne um local mais usável por todos, indiscriminadamente.

Abstract

Due to the fast growing of the Web in the last decades and to the interactivity brought by the Web 2.0, government services, entertaining, and education are, more and more, made available in the Internet. However, part of the efforts that follow this growing does not consider the differences present in the whole population of users.

There are approaches and techniques aiming at the inclusive design in phases of analysis, design, development, and evaluation of *websites*. In the initial stages it is possible to identify a number of tasks and different ways of using a user interface (UI), but questions related to different user needs may arise in different contexts of use not foreseen in stages occurring previously than real usage.

Self adaptive UIs represent a promising approach on promoting the accessibility and usability in the Web and hence making the needs originated in the most varied contexts of use to be considered not only in the analysis, in the project, or in the implementation, but also during the UI usage. In this regard, the research question that guided the study proposed in this thesis was: Is it possible to develop a continuous evaluation – based on detailed usage data – that generates UI adjustments aiming at the Design for All?

This thesis aimed at presenting existing approaches on website evaluation and proposing a new approach that uses client-side event logs as data source, identifies users' behavior, and adjusts the UI automatically according to the interactions used. Since the presented context raises questions involving technical and social issues related to the development of information systems, the main theoretical-methodological basis followed is the Organizational Semiotics.

The main objective of the thesis was to support continuous evaluation of websites, aiming at eliminating accessibility barriers and usability problems. The results obtained in this thesis are:

1. formalization of the heuristics that supports the identification of usage incidents and adjustment analysis; the heuristics obtained approximately 65% of effectiveness;

2. definition of a set of 22 requirements for evaluation tools, based on a study involving 11 users of an Inclusive Social Network;

3. design and redesign of the report presented to evaluators, counting on the participation of 28 information technology specialists;

4. revision of a taxonomy to classify evaluation tools considering 4 dimensions (i.e., interaction between participant and evaluator, data source, effort level related to the evaluators and participants, and automation type supported by the tool);

5. identification of statistically significant differences involving length ($p = 0.01$) and presence of mouse ($p = 0.01$) in the event streams related to sessions coming from assistive technology users or from users that do not use assistive technology; in addition, these results allowed the correct classification of these event streams in more than 80% of cases;

6. identification of patterns related to how assistive technology users interact with websites in real context of usage;

7. presentation of the viability of applying a norm-based approach to automatically generate and apply adjustments, considering usage data collected in a real context of use.

From the results presented in this thesis, one expects to support self tailoring approaches of websites that are based on the analysis of client-side event logs and use Web Usage Mining techniques, considering differences and contributing for the Web to become an environment more usable by all, indiscriminately.

Agradecimentos

Primeiramente, gostaria de agradecer à professora Maria Cecília Calani Baranauskas por ter acreditado no projeto e por todos os ensinamentos e conselhos que certamente levarei para muito além do doutorado.

À minha família que sempre foi meu porto seguro e sempre me deu forças. À minha companheira Rosimeire de Oliveira com quem dividi minhas angústias e felicidades por mais de uma década, incluindo tudo que veio desde antes do meu ingresso na graduação até a conclusão desta tese. Ao meu irmão Douglas que sempre foi uma inspiração e referência para mim, desde quando eu era moleque. À minha mãe Vanilda pelos valores ensinados e por representar, em pessoa, o fato de que com muito trabalho conseguimos alcançar nossos objetivos, mesmo quando as coisas parecem estar contra nós. À minha madrinha Vandete e ao meu padrinho João por terem me apoiado em diversos momentos da minha vida.

Ao Antonio Graeff, gerente de operações na época em que trabalhei como *webmaster* na Folha Online, por ter me ensinado muito do que sei hoje sobre tecnologia e pelo enorme apoio dado quando cursava graduação e trabalhava sob seu comando. Sem seu apoio e seus ensinamentos dificilmente estaria escrevendo agora, após anos de estudo, a minha tese de doutorado. E ao Rogério Cassimiro pelos ensinamentos na época que foi meu chefe na Folha Online e por ter liberado a minha transferência da fotografia para tecnologia da informação.

Ao amigo José Ricardo Jayme pela amizade que já data de algumas décadas e por todas as forças que me deu.

Aos amigos Hamilton Matos e Gabriela Doratioto que nem sempre são tão presentes como quando estávamos na graduação, mas que sei poder contar sempre com eles.

Ao amigo Leonardo de Cunha Miranda pela troca de experiências, pelas discussões e bate-papos sempre riquíssimos na sala 90, fonte de muito conselho, apoio e ideias novas.

Ao amigo Leonelo Dell Anhol Almeida pela parceria em tantos projetos ao longo do mestrado e do doutorado e pelas maratonas de escrita e programação nas quais muitas vezes tivemos *insights* valiosíssimos.

Aos amigos Ricardo, Lucas, Tiago, Wellington e Sabrina, da Pensão dos Ratões, local onde “morava” durante a semana nos períodos em que eu cursava disciplinas, com quem as trocas de experiências sempre trouxeram alento e os *happy hours* sempre resgatavam um pouco da alegria que muitas vezes parecia estar distante.

Aos professores Ismar Frango e Pollyana Mustaro, da Universidade Presbiteriana Mackenzie, por me encorajarem quando me propus a me inscrever no programa de pós-graduação do Instituto de Computação da UNICAMP.

Ao professor Fabio Paternò pela oportunidade de realizar o sonho de visitar o laboratório *Human Interfaces in Information Systems*, sediado no *Consiglio Nazionale delle Ricerche* (CNR), em Pisa, Itália. Dado que desde quando comecei a estudar sobre avaliação de Interfaces de Usuário e prospectar tópicos de pesquisa sempre acabava me deparando com artigos produzidos por ele e sua equipe; assim, sempre imaginei poder trabalhar com sua equipe. Ao Tonio Carta, ao Claudio Porta e ao Federico Bellucci pela receptividade e por terem me ajudado em diversas ocasiões enquanto estava visitando o HIIS/CNR.

Aos grupos de pesquisa Todos Nós, e-Cidadania e EcoWeb, dos quais fiz/faço parte e que sempre me orgulho de estar no meio de tantos “feras” da área de Interação Humano-Computador.

Ao Núcleo de Informática Aplicada à Educação (NIED/UNICAMP), ao Instituto de Computação (IC/UNICAMP) e o ao Laboratório de Acessibilidade (LAB/UNICAMP) pela infraestrutura e apoio em diversas das fases desta tese.

À FAPESP por ter acreditado no projeto de pesquisa.

Contents

List of Tables

List of Figures

Chapter 1

Introdução

A Web atualmente conta com cerca de 665 milhões de *websites*[1] e em levantamento feito durante a tese foi identificado que 96% dos *websites* que mais geram tráfego no mundo contam com barreiras de acessibilidade de fácil identificação e correção. Estes dados indicam que o atual cenário relacionado à acessibilidade na Web requer estudos e desenvolvimento de aplicações que auxiliem na análise, identificação e correção de problemas. Essas ações se fazem necessárias e são fundamentais para que este cenário seja mudado. Dessa forma, esta tese de doutorado é uma ação para a remoção de barreiras de acessibilidade e problemas de usabilidade, considerando o Design para Todos como norte.

Acessibilidade na Web significa que pessoas com diferentes tipos de limitação podem perceber, entender, navegar, interagir e contribuir para a Web. Barreira de acessibilidade é qualquer coisa que dificulte ou impossibilite pessoas com deficiência de usar a Web [125]. Usabilidade, em suma, é a capacidade de um produto ser utilizado por usuários específicos para atingir objetivos com eficiência e satisfação, dentro de um contexto de uso. Ainda, contexto de uso envolve os usuários, tarefas, equipamentos (hardware, software e materiais), ambiente físico e social em que o produto é usado [58]. Problema de usabilidade pode ser definido como aspectos de IU que reduzem a usabilidade do sistema para usuários [92].

Associado à importância do acesso às Tecnologias de Informação e Comunicação (TICs) está o entendimento de como essas tecnologias são utilizadas por todos seus usuários, o que remete à combinação dos conceitos acessibilidade e usabilidade (A&U), uma vez que questões relacionadas a uma das disciplinas pode contribuir para a outra e vice versa (e.g., teclas de atalho para regiões da tela aumentam a acessibilidade para usuários que usam leitores de tela e também auxiliam usuários que preferem usar atalhos para aumentar eficiência). Segundo Abascal e Nicolle [4], se serviços não são acessíveis, eles são inúteis para pessoas com deficiências; se serviços são acessíveis, também é impor-

[1] `http://news.netcraft.com/archives/2012/07/03/july-2012-web-server-survey.html`

tante que usuários consigam executar tarefas com sucesso, facilmente e eficientemente. Os autores também comentam que designs ruins não são difíceis de usar somente para pessoas com deficiência, mas para todos. Dessa forma, apoiar a acessibilidade na Web não exclui nenhum usuário e estende o conceito de usabilidade [101].

A avaliação de interfaces de usuário (IU) é uma forma de identificar problemas no projeto de *websites*. Complementarmente, dado o volume de dados resultantes de avaliações de IU, o uso de ferramentas de avaliação se faz necessário para liberar avaliadores de tarefas custosas e que consomem muito tempo, como registro manual de ações ou análise manual de *logs*.

Há diferentes formas de avaliar uma página Web. É possível avaliar o código fonte tendo em vista validar a marcação utilizada, avaliar o conteúdo de uma página Web considerando a adequação do conteúdo ao público-alvo e analisar os dados de uso, que refletem como uma IU é utilizada. Essas formas de avaliar uma página Web podem envolver tanto a remoção de barreiras de acessibilidade quanto de problemas de usabilidade.

A avaliação de código fonte é fundamental para que a página Web siga os padrões e definições das tecnologias em uso (e.g., HyperText Markup Language, JavaScript, Cascading Style Sheets). Este tipo de avaliação é fundamental para que as páginas sejam compatíveis entre diferentes navegadores e dispositivos. A avaliação de conteúdo pode envolver diferentes técnicas para melhorar informação textual, tornar conteúdo acessível às pessoas com diferentes níveis de letramento, sumarização de conteúdo, entre outros. Por fim, a avaliação de dados de uso envolve o estudo do comportamento dos usuários ao interagir com o sistema. Esse comportamento é refletido em *logs* que indicam como uma página Web é utilizada. Os dados de uso são uma fonte rica de dados porque possibilitam a análise do contexto de uso real. Este fato é de grande valia especialmente quando estudos envolvem acessibilidade, uma vez que a reprodução de contextos de uso reais envolvendo tecnologias assistivas (TA) em ambientes controlados é um desafio à parte.

Há diferentes métodos de inspeção de IU. Inspeção pode ser definida como o conjunto de métodos baseados em se ter avaliadores examinando aspectos relacionados à usabilidade de uma IU. Exemplos de métodos de inspeção são: Inspeção Heurística, em que o avaliador analisa a interface de usuário considerando uma pequena lista de heurísticas de usabilidade; Revisão de Diretrizes, em que a IU é verificada de acordo com um conjunto de diretrizes de usabilidade; Inspeção de Consistência, em que o avaliador verifica a consistência de todos os elementos da IU em avaliação considerando uma família de outras IUs; Percurso Cognitivo, em que o avaliador simula os caminhos que os usuários farão na UI para executar as tarefas e assim identificar eventuais problemas de usabilidade [92].

Situando a utilização de dados de uso de maneira complementar aos métodos de inspeção apresentados tem-se que a avaliação de dados de uso é uma forma de verificar quais são as ações executadas realmente, o que pode servir de fonte de dados para uma inspeção usando Percurso Cognitivo. Complementarmente, padrões identificados nos dados de uso podem ser contrastados com heurísticas ou diretrizes de usabilidade. Por fim, dados de uso também permitem aferir se as IUs são usadas de maneira consistente pelos usuários através da análise de padrões.

1.1 Contexto e objetivo

Esta tese foi construída sobre resultados da dissertação de mestrado em Ciência da Computação do autor deste documento. O trabalho de mestrado resultou em um modelo de avaliação e uma ferramenta de avaliação chamada WELFIT (*Web Event Logger and Flow Identification Tool*), que implementa esse modelo [100]. Na dissertação de mestrado a ferramenta proposta tinha como objetivo capturar dados de uso em contexto de uso real e apontar possíveis incidentes de uso. O incidente de uso considerado nesta tese tem raiz no termo incidente crítico que é uma técnica apresentada por Flanagan [39] e, em Interação Humano-Computador (IHC), foi usado por Hartson et al. [48] em um dos primeiros trabalhos que apresentam resultados comparativos entre avaliações de usabilidade local e remota. Naquele trabalho Hartson et al. [48] usam a expressão incidente crítico para representar um problema de usabilidade, funcionalidade faltante e outras formas em que um sistema falha ao não se adequar às necessidades dos usuários. Dessa forma, neste trabalho revisitamos o termo incidente crítico usado em avaliações remotas de usabilidade como em [48, 19] e propomos o termo incidente de uso como sendo algo que representa um incidente crítico ou uma barreira de acessibilidade.

Nesta tese a ferramenta WELFIT foi estendida com o intuito de ajustar a IU de acordo com padrões de comportamento dos usuários e foi nomeada WELFIT 2.0; *logs* do lado do cliente e técnicas de *Web Usage Mining* (WUM) são combinadas tendo em vista o ajuste automático de IU e o Design para Todos como meta principal.

Este projeto segue a visão proposta pelo Design Universal, também conhecido como Design para Todos. Connell et al. [24] apresentam que essa visão tem como objetivo o design de produtos e ambientes que sejam usados por todos, na maior extensão possível, sem a necessidade design especializado.

Dessa forma, as soluções propostas e os objetivos almejados nesta tese têm sempre como propósito considerar todas as pessoas, independentemente de eventuais limitações físicas ou funcionais. Assim, quando técnicas de WUM são utilizadas para formar grupos, consideram-se as ações utilizadas (representadas por meio de eventos disparados) em vez de considerar qual deficiência ou quais características os usuários possuem. Se pessoas (com e sem deficiência) utilizam uma determinada IU da mesma maneira e podem se beneficiar do mesmo ajuste, então a ferramenta proposta assim o fará.

A automatização de ferramentas de avaliação pode envolver: captura (i.e., gravação dos dados de utilização), análise (i.e., identificação de problemas) e crítica (i.e., sugestões de como melhorar a IU avaliada) [59]. Complementarmente, pode envolver ajuste para eliminar problemas identificados [100]. O levantamento de literatura realizado identificou duas principais lacunas em relação à automatização de ferramentas de avaliação de IU baseadas em dados de uso: crítica e ajuste. Crítica no que se refere às ferramentas que apresentam propostas de soluções para os problemas identificados. Ajuste quando a ferramenta tenta eliminar os problemas identificados através do ajuste da IU avaliada. Dessa forma, esta tese aborda a lacuna identificada em relação ao ajuste, experimentando ajustes automaticamente nas IUs avaliadas. O termo "experimentar" é utilizado nesta tese porque a ferramenta cria continuamente diferentes experimentos relacionados aos ajustes; este tópico será abordado em detalhes no capítulo 6.

Esta tese é norteada pela seguinte pergunta de pesquisa: é possível desenvolver uma avaliação contínua, baseada em dados detalhados de uso, que gere ajustes à IU tendo como objetivo o Design para Todos? Desta forma, o objetivo deste trabalho foi apresentar como uma ferramenta computacional pode capturar dados de uso e utilizá-los para ajustar uma IU, tendo como base a forma como os usuários reais utilizam *websites* reais em contexto e ambiente reais. Este objetivo foi subdividido em etapas que organizam as contribuições apresentadas nos capítulos que seguem e compõem esta tese. Em linhas gerais, estas etapas apresentam a proposta de sumarização de dados de uso, o levantamento de requisitos, como se dá a resposta ao convite à avaliação, a efetividade da sumarização proposta quando analisada por potenciais avaliadores de IU, a apresentação do sistema computacional proposto como ferramenta de estudo de *logs* de utilização e a proposta de abordagem de IU autoajustáveis.

1.2 Referencial teórico

O principal referencial teórico utilizado no trabalho é a Semiótica Organizacional (SO) [115]. SO é uma disciplina que lida com informação e sistemas de informação de uma maneira que leva em consideração tanto questões técnicas quanto aspectos sociais e humanos [117]. A SO conta com um conjunto de métodos chamado MEASUR (*Methods for Eliciting, Analyzing and Specifying Users' Requirements*), que pode ser utilizado para o entendimento, desenvolvimento, gerenciamento e uso de sistemas de informação. O MEASUR conta com 5 métodos principais, são eles [62]:

- *Problem Articulation Method* (PAM), que auxilia na clarificação do problema tratado;
- *Semantic Analysis Method*, que apoia o levantamento e representação de requisitos;
- *Norm Analysis Method* (NAM), que fornece meios para especificar padrões gerais de comportamento dos agentes dentro do sistema;

- *Communication and Control Analysis*, que auxilia na análise de toda comunicação ocorrida entre todos os agentes e sistemas identificados no PAM;

- *Meta-Systems Analysis*, que apoia a solução de metaproblemas no planejamento e gerenciamento do projeto.

Neste trabalho, o artefato do MEASUR utilizado é a Escada Semiótica. Ela apoia a análise de sistemas de informação em seis diferentes degraus, contribuindo para a clarificação do que é necessário produzir para que um sistema resolva não somente os problemas relacionados à plataforma de Tecnologia da Informação (TI), mas também considere aspectos sociais da utilização deste sistema [116]. Ela foi usada para organizar as características e definir os requisitos para ferramentas de avaliação baseadas em *logs* de eventos, assim como apoiar a especificação do modelo em desenvolvimento. Os requisitos definidos em [103] foram revisitados e são apresentados no capítulo 3.

Em suma, o PAM foi aplicado na articulação do problema estudado assim como na definição dos requisitos do sistema e o NAM foi aplicado para estabelecer o modelo semântico do sistema autoajustável proposto. Dessa forma, uma vez que os padrões de comportamento são identificados, uma condição para selecionar ocorrências desses padrões dentro das cadeias de eventos é definida (seletor). Portanto, assim que um padrão é encontrado, um experimento para o ajuste é criado. O ajuste relacionado ao experimento combina os elementos de IU identificados pelo seletor com o ajuste abstrato (i.e., que não referencia nenhum elemento concreto de IU) definido na norma. Assim, a troca de um elemento de IU abstrato por um elemento concreto identificado pelo seletor resulta em um ajuste concreto.

No contexto de avaliação de IU, o referencial da SO desempenha um papel importante, uma vez que o modelo em desenvolvimento deve considerar tanto questões relacionadas à infraestrutura computacional quanto questões centradas nos usuários e nos avaliadores, abordando os conceitos de acessibilidade e usabilidade.

O referencial para os projetos de experimento do trabalho é baseado em [126]. O WELFIT segue o processo de *Web Usage Mining apresentado* em [88], passando pelas etapas de captura de dados, pré-processamento (e.g., filtragem de dados, identificação de usuários, identificação de sessões), descoberta de padrões (e.g., agrupamento, classificação, *sequence mining*) e pós-processamento de conhecimento (e.g., visualização, relatórios, personalização).

A tese faz uso da heurística proposta na dissertação de mestrado [100] e é formalizada e avaliada no capítulo 2 desta tese. A heurística é fundamental para identificar prováveis incidentes de uso e para analisar se os ajustes foram bem sucedidos. Complementarmente, o WELFIT foi remodelado no que diz respeito ao modelo de dados, algoritmos de agrupamento e captura. A alteração na base de dados foi necessária para dar suporte à criação de ajustes e avaliação dos mesmos. Por fim, a implementação do algoritmo de agrupamento hierárquico *Average Linkage* foi fundamental para reunir sessões que representam comportamentos semelhantes e, consequentemente, devem receber os mesmos ajustes. O módulo de captura foi alterado tendo em vista torná-lo mais compatível entre os diferentes navegadores considerando bibliotecas de programação JavaScript. Outro aspecto relacionado ao uso de bibliotecas de programação de JavaScript é facilitar a criação de ajustes, uma vez que facilitam a manipulação de aparência e estrutura dos elementos de uma página Web. Por fim, este aperfeiçoamento no módulo cliente tem relação direta com a experiência obtida durante período de visita ao laboratório *Human Interfaces in Information Systems* (HIIS), no *Consiglio Nazionale delle Ricerche* (CNR), em Pisa, Itália.

O conceito de *tailoring* seguido neste trabalho é o definido por Mørch [71]. Segundo o autor, *tailoring* é a adaptação de sistemas de informação para práticas específicas de desenvolvedores, usuários ou grupo de usuários. Ele ainda apresenta três níveis de *tailoring*:

1. Customização: Modificar a apresentação de objetos e seus atributos através da seleção de valores predefinidos;

2. Integração: Criação ou gravação de sequência de ações que resultam em nova funcionalidade, armazenada na aplicação como um componente ou como um comando;

3. Extensão: Melhorar a funcionalidade de uma aplicação através da inserção de novo código.

Em suma, o referencial apresentado apoia o WELFIT 2.0 no uso de dados detalhados de uso, tendo sua fundação na SO tanto no desenvolvimento como na concepção de uma abordagem autoajustável para aplicar *tailoring* de nível 3.

1.3 Contribuições e organização

As principais contribuições desta tese são:

- Formalização da estrutura de dados para representar o uso por meio de um grafo direcionado (grafo de uso);

- Revisitação dos requisitos para ferramentas de avaliação de *websites* baseadas em *logs* de eventos, apresentados em uma instância da Escada Semiótica;

- Avaliação do impacto do convite à avaliação feito a usuários em um contexto de exclusão digital, com grupo de 11 participantes;

- Criação e disponibilização de um conjunto de signos representando eventos de IU;
- Avaliação do entendimento que pessoas com perfil de avaliadores têm do grafo de uso e dos signos nele contidos, com grupo de 28 participantes com perfil de avaliadores;
- Análise de padrões e de estrutura das cadeias de eventos, considerando 246 sessões e 15 meses de dados observacionais;
- Revisão do estado da arte relacionado à avaliação de *websites*;
- Proposta de uma taxonomia para classificar ferramentas de avaliação;
- Demonstração da viabilidade de se aplicar uma abordagem de ajuste automático considerando dados detalhados de uso, sem depender de modelos de tarefa nem de manutenção de gramáticas, com base em 30 meses de dados observacionais.

Os capítulos desta tese apresentam diferentes etapas do desenvolvimento desta nova versão do WELFIT. Cada capítulo conta com uma contribuição do trabalho e contém texto integral de artigos publicados em conferências ou submetidos para revistas. A Figura 1.1 apresenta o relacionamento entre os próximos 5 capítulos e os 3 apêndices desta tese; as arestas representam a relação entre os trabalhos.

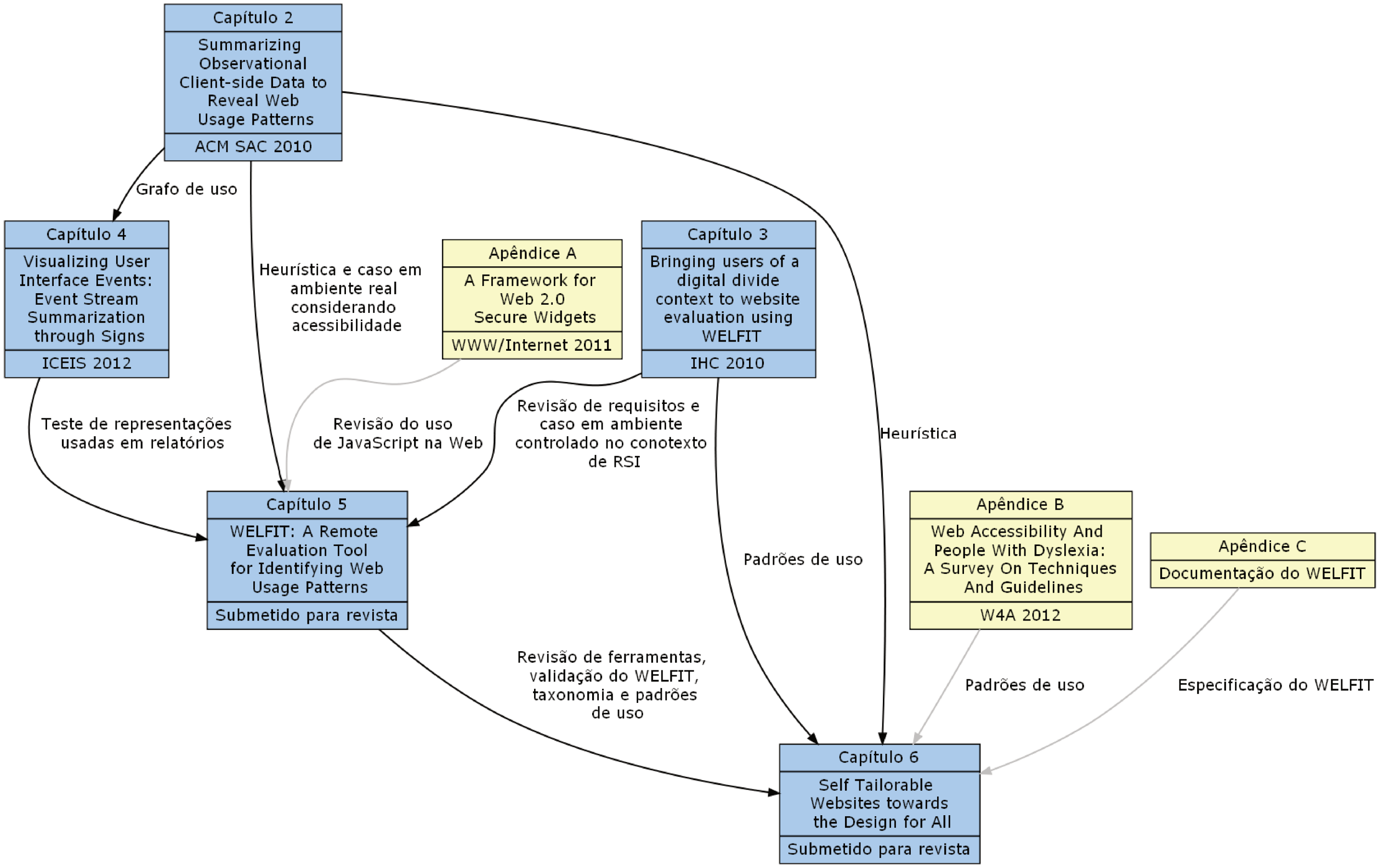

Capítulo 2
Summarizing Observational Client-side Data to Reveal Web Usage Patterns
ACM SAC 2010
Grafo de uso
Capítulo 4
Visualizing User Interface Events: Event Stream Summarization through Signs
ICEIS 2012
Heurística e caso em ambiente real considerando acessibilidade
Apêndice A
A Framework for Web 2.0 Secure Widgets
WWW/Internet 2011
Capítulo 3
Bringing users of a digital divide context to website evaluation using WELFIT
IHC 2010
Teste de representações usadas em relatórios
Revisão do uso de JavaScript na Web
Revisão de requisitos e caso em ambiente controlado no conotexto de RSI
Heurística
Capítulo 5
WELFIT: A Remote Evaluation Tool for Identifying Web Usage Patterns
Submetido para revista
Padrões de uso
Apêndice B
Web Accessibility And People With Dyslexia: A Survey On Techniques And Guidelines
W4A 2012
Apêndice C
Documentação do WELFIT
Revisão de ferramentas, validação do WELFIT, taxonomia e padrões de uso
Padrões de uso
Especificação do WELFIT
Capítulo 6
Self Tailorable Websites towards the Design for All
Submetido para revista

Figure 1.1: Mapeamento entre trabalhos que compõem a tese e como eles se relacionam.

Dessa forma, o corpo da tese será composto pelos seguintes capítulos:

2) *Summarizing Observational Client-side Data to Reveal Web Usage Patterns*, que consolida ideias propostas para sumarizar dados observacionais coletados no lado do cliente. Foi apresentado oralmente em março de 2010 durante o *25th ACM Symposium On Applied Computing*, em Sierre, Suíça. O trabalho formaliza a construção dos grafos de uso e avalia a heurística usada para identificar incidentes de uso em *logs* de uso;

3) *Bringing Users of a Digital Divide Context to Website Evaluation Using WELFIT*, que considera avaliação de websites em contexto de diversidade, desenvolvido no âmbito do projeto e-Cidadania[2]. Foi apresentado oralmente em outubro de 2010 no IX Simpósio de Fatores Humanos em Sistemas Computacionais (IHC 2010), em Belo Horizonte. O trabalho apresenta a revisão dos requisitos de ferramentas de avaliação, considerando o uso da ferramenta sob a ótica das pessoas que são convidadas à avaliação;

4) *VISUALIZING USER INTERFACE EVENTS: Event Stream Summarization Through Signs*, que conta com estudo sobre como avaliadores interpretam resultados do WELFIT. Foi apresentado oralmente em julho de 2012 no *14th International Conference on Enterprise Information Systems* (ICEIS 2012), em Wroclaw, Polônia. O trabalho detalha o design e a avaliação dos signos utilizados nos relatórios gerados pelo sistema WELFIT, considerando o uso da ferramenta sob a ótica dos avaliadores;

5) *WELFIT: A Remote Evaluation Tool for Identifying Web Usage Patterns through Client-Side Logging*, que consolida os dois primeiros anos do projeto de tese. Conta com a visão obtida durante o desenvolvimento da ferramenta de base (no mestrado) e com a evolução e o levantamento bibliográfico obtidos nos primeiros anos do desenvolvimento da tese. Esse trabalho, submetido para publicação em revista científica, representa a síntese de resultados essenciais para o desenvolvimento da pesquisa. O trabalho é a base para que o objetivo principal da tese em desenvolvimento seja alcançado: possibilitar através do uso de *logs* do lado do cliente o autoajuste de *layout* e estrutura de *websites*;

6) *Self Tailorable Websites towards the Design for All*, que traz detalhes da abordagem proposta para possibilitar ajuste automático de *layout* e estrutura de *websites* a partir da utilização de *logs* de eventos. O trabalho, submetido para publicação em revista científica, detalha a proposta de avaliação continuada de *websites* tendo em vista a eliminação de barreiras de acessibilidade e problemas de usabilidade.

[2] `http://styx.nied.unicamp.br:8080/ecidadania`

Em suma, os trabalhos que compõem o corpo da tese apresentam e discutem: a heurística utilizada para indicar incidentes de uso; revisão de requisitos e como usuários interagem com o convite à participação de avaliação; como avaliadores utilizam e interpretam os grafos de uso gerados e os signos neles contidos; revisão da literatura indicando soluções e limitações que corroboram o projeto de tese; por fim, automatização dos ajustes, fundamentada nos outros resultados para apresentar a principal contribuição da tese.

Complementarmente, a tese conta com outros trabalhos e documentos na seção de apêndices. Esses trabalhos tratam de temas que contribuíram direta ou indiretamente para o desenvolvimento do projeto e complementam a leitura do corpo da tese. Dessa forma, eles não fazem parte do corpo principal, mas apresentam estudos relacionados desenvolvidos no decorrer da tese. A seção de apêndices é composta pelos seguintes trabalhos:

A) *A Framework for Web 2.0 Secure Widgets*, que discute questões relativas à segurança de troca de dados entre *websites*. O trabalho foi apresentado em novembro de 2011 durante o WWW/Internet 2011, no Rio de Janeiro. O artigo conta com levantamento de questões técnicas em relação à captura de informações no lado do cliente e é a base para o rationale de como a captura e transmissão de dados é feita no WELFIT;

B) *Web Accessibility and People with Dyslexia: A Survey on Techniques and Guidelines*, que apresenta levantamento de diretrizes e técnicas envolvendo o uso da Web por pessoas com dislexia. O trabalho foi apresentado em abril de 2012 durante o *9th International Cross-Disciplinary Conference on Web Accessibility* (W4A 2012), em Lyon, França. O levantamento, que conta com resultados relacionando problemas e soluções, foi uma das fontes consideradas na definição das normas de ajuste, fomentando a última etapa da tese;

C) Documentação do WELFIT, que detalha os casos de uso, base de dados e fluxograma do funcionamento geral da ferramenta.

Em linhas gerais os apêndices detalham: questões de segurança a serem consideradas tanto na captura quanto no ajuste; levantamento de problemas, soluções e diretrizes, que servem de base para definição de normas usadas em ajustes; documentação da ferramenta.

Cada capítulo/apêndice apresenta uma etapa da pesquisa. No decorrer desta tese os capítulos/apêndices que serviram de base ou contribuíram para o desenvolvimento de um capítulo seguinte serão referenciados. Considerando a presente estrutura e organização desta tese, a quantidade de dados capturados e mencionados ao longo dos capítulos aumenta de acordo com a etapa em que cada trabalho foi escrito. O leitor também poderá verificar que, ao longo dos capítulos, as definições e abordagens apresentadas evoluem ao longo do trabalho, refletindo o desenvolvimento da pesquisa e o aprofundamento no tópico estudado.

Chapter 2

Summarizing Observational Client-side Data to Reveal Web Usage Patterns

Vagner Figuerêdo de Santana
Institute of Computing-UNICAMP
1251 Albert Einstein Av., Campinas, SP, Brazil
+55 (19) 3521-5838
vsantana@ic.unicamp.br

M. Cecília C. Baranauskas
Institute of Computing-UNICAMP
1251 Albert Einstein Av., Campinas, SP, Brazil
+55 (19) 3788-5870
cecilia@ic.unicamp.br

Abstract: Client-side event *logs* may reveal patterns of usage of Web pages. Nevertheless, extracting useful and novel information from this voluminous data set is a challenge for evaluation tools, since a few minutes simple task may result in a sequence of hundreds of events. This work contributes with a technique to process these logs and build a Web page's usage graph summarizing statistical information of the Web page usage concerning one or more sessions. This graph reveals patterns of real usage data, which Human-Computer Interaction specialists may find useful for inspecting accessibility and usability issues. Moreover, Web usage miners can reuse the usage graph to apply other techniques to discover other patterns or rules.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces – Evaluation/methodology. D.2.2 [Software Engineering]: Design Tools and Techniques – User interfaces.

General Terms: Design, Human Factors, Verification.

Keywords: Usage Patterns, Websites Evaluation Tool, Client-side event logs.

2.1 Introduction

Several studies involving observational log data have been conducted by website evaluation tools and WUM (Web Usage Mining) algorithms, but most of them make use of server logs as data source. This trend emerges from the fact that server logs are a natural product of Web servers functioning, so the cost of capturing this data is straightforward. However, server logs do not provide detailed information about what users do when they are interacting with the user interface (UI) elements of a Web page in a granularity level higher than the page-view level [7]. This fact occurs because Web server logs keep track of requested pages, HyperText Transfer Protocol (HTTP) method used, time of the request, etc.

Client-side events represent how users interact with Web pages. Since it is possible to capture and record them, the resulting logs allow keeping track of all action performed by users (e.g. mouse movements, mouse clicks, or pressed keys) or by the browser (e.g. image is loaded, form automatically receives focus, or some error occurs). Recording this event stream allows one to rebuild all actions performed within the UI. Nevertheless, due to detailed information provided by client-side logs, the amount of data grows in an impressive manner if compared to server-side data.

In the Web context, accessibility barrier is anything that makes it difficult or impossible for people with disabilities to use the Web [125]. Usability problems can be defined as UI aspects that reduce the system's usability for end users [92]. In some cases, both issues can occur in a context of use. For example, if a Web page has a link that is too small and then too hard to be pointed out by users with disability and users without disabilities as well, it represents an accessibility barrier and a usability problem. Bringing this example to evaluation tools based on client-side event logs, we can look for repetitions of mouse events over an element before performing a successful click, which may be a usage pattern.

Usage pattern can be defined as a set of combinations of events representing the behavior of one or more users. Thus, Web usage patterns show how users interact with specific Web page elements. They represent an interesting source of information, opening new possibilities of scenarios regarding how to evaluate UI design and how to adjust the

UI as well. The scenario we will focus in this work refers to the use of client-side logs, remotely, and during real use. This scenario differs from tests in controlled environments, which are artificial and may influence results [96]. Also, the real scenario of use plays an important role when dealing with accessibility due to difficulties in replicating the diversity of configuration of hardware and software used by participants [13].

This work is organized as follows: section 2.2 presents existing approaches to websites evaluation tools, section 2.3 presents how the client-side data used in this work was captured; section 2.4 discusses characteristics of client-side event data and how to build the usage graph; section 2.5 presents how the usage patterns are revealed; section 2.6 presents the results achieved so far; and, finally, section 2.7 presents the discussion and future directions.

2.2 Website Evaluation Tools

When evaluating a website's UI, the participant can make informal use of it (i.e., the evaluation requires the completion of freely chosen tasks) or formal use (i.e., the evaluation requires the completion of specifically selected tasks) [59]. Tools can involve capture (i.e., logging usage data), analysis (i.e., identification of problems), and critique (i.e., suggestion of improvements) [59].

WebVIP is one of the first efforts to apply automatic capture of client-side events through the use of JavaScript. The vocabulary of events, which stands for the number of different event types, is restricted [77]. The main drawback of the tool is the environment configuration, a set of steps required to set up the tool; it involves the use of a local copy of the entire website being evaluated.

WET is a JavaScript logger for formal tests. It uses cookies to store logged data, influencing the reduction of the vocabulary of events due to available space issues. The user must indicate the starting and ending of the capture through tool's controls so that the logs can be associated with the test session [37].

WebRemUSINE is a tool that makes automatic capture and analysis of websites interaction logs in order to detect usability problems through remote evaluation. The log analysis is based on the comparison between the paths made by users and the optimum task model previously configured. In addition, the user must select the tasks s/he is performing so that the events captured can be related to the task [83]. The drawbacks are related to:

a) the environment configuration involving the definition and specification of optimum task models,

b) the reduction of useful space due to the display of the task list,

c) the requirement to select the task s/he is performing, and

d) the need of a Java plug-in.

WebQuilt is an automatic capture and analysis tool that uses page-view level logs as data source. It uses a proxy-logger that mediates between users and Web servers and stores the communication between them. This approach prevents changes in the client-side, and enables the identification of a large number of people navigating between different websites. Usability tests are possible on any website, even if the evaluators are not the maintainers of the websites visited by the participants [52]. The drawbacks of the tool are related to:

a) the use of server logs and

b) all URLs used in the evaluated pages are changed to point to the proxy-logger.

MouseTrack is a proxy-based usability evaluation system capable of automatic event capture and data analysis. The tool fetches the Web pages being evaluated and modifies them by inserting JavaScript code responsible for capturing mouse movements. It also provides an online configuration and visualization tool that shows the mouse path followed by website's visitors [7]. The tool combines interesting approaches when uses the proxy-logger model and JavaScript to capture events at client-side. However, the tool focuses only on mouse movements, which represents a restriction when dealing with accessibility.

UsaProxy is a proxy-based usability evaluation system that performs automatic capture and analysis of client-side events. It uses JavaScript and focus on usability tests [8]. The tool brings important points dealing with the willingness of users to take part in remote usability evaluation. However, the limitation of the tool is the environment configuration, which requires reconfiguration of the user's browser or a proxy setup.

WebinSitu is a proxy-based tool, an enhanced version of UsaProxy that focuses on the comparison of the behavior of blind and sighted users [13]. The tool's authors also stress the importance of capturing real life data especially when dealing with accessibility, since realistic studies with blind users are difficult to conduct in the lab due to difficulties in replicating the diversity of assistive technology and configuration normally used by participants [13]. The tool inherits the drawbacks of UsaProxy.

Web Utilization Miner is a miner system for the discovery of navigation patterns in websites using server-logs as data source. In addition, it uses directed graphs to represent the patterns [113]. WebSIFT (Web Site Information Filter) is a miner system that uses server-side data and count on algorithms to discover knowledge dealing with usage, content, and structure [25]. LumberJack is a tool that processes Web server logs and uses the content and hyperlinks to build a model of user activity. It applies clustering analysis and then computes a number of statistical analysis for each discovered group [21]. DCW (*Descubridor de Conhecimento en la Web*) is a tool that uses Web Usage Mining over server-side logs in order to discover navigation patterns and association rules [35].

Web Utilization Miner [113], WebSIFT [25], LumberJack [21], and DCW [35] represent interesting contributions regarding the problems of evaluating Web usage by applying statistical analysis to identify usage patterns, and association rules. Other important contribution is their focus on how to represent the outcomes in a summarized way, revealing clusters of users and their common behaviors. However, the logs commonly used in these studies are captured at server-side, leaving higher granularity interactions uncovered. Thus, this work presents a model that facilitates the environment configuration, grants control to the user, returns summarized information of captured sessions, and also provides the tool as a service available in the Web so that evaluators can reuse it with minimum effort.

2.3 The WELFIT Approach

This work is part of the project of a tool called WELFIT (Web Event Logger and Flow Identification Tool). The project involves the development of a model to capture client-side event logs, transfer them to a server, and mine the captured logs in order to discover usage patterns in Web pages. In addition, the tool aims at revealing accessibility barriers and/or usability problems that assistive technology users may have faced.

The tool considers the requirements of website evaluation tools defined in [103]. The requirements highlight some characteristics of evaluation tools based on event logs, namely:

1. The capture should be lightweight and should not interfere with the website use;
2. The user must be aware of the evaluation, having control of the capture;
3. The user must not feel disturbed by the system. WELFIT's capture module follows the model defined in [102].

WELFIT captures all types of standard JavaScript client-side events remotely, during real use, when users are in their natural usage environment. Considering all types of events is essential, since each one represents one kind of action performed by the user or by the browser, thus allowing the analysis and the reconstruction of everything that happened during the real use.

The WELFIT environment configuration has three steps:

1. The website administrator must register him/herself at WELFIT's Web administrative interface;

2. Once logged, the administrator can register the websites s/he wants to evaluate;

3. Once the website is registered, s/he includes the call to the JavaScript client module in all website's pages to be evaluated.

The data captured by WELFIT is performed according to the following steps:

1. When a user accesses a website with a reference for the tool's client module, the server then verifies if the request comes from a registered website;

2. If the website is registered, then as soon as the tool's client module is loaded at client-side, the user receives an invitation to be part of the evaluation;

3. The user is also asked if s/he is using an assistive technology. This information is used to mark the session and to analyze the behavior of groups of users sessions;

4. After submitting the answers, if the user agreed to participate, then the capture starts and the events captured at the client-side are sent via asynchronous communication to the WELFIT's Web server. Thus, as soon as some data is sent to the tool's Web server and recorded into the tool's database, the evaluator can then login and check the resulting usage graph.

2.4 Representing Client-side Logs

Graphs are the most common form of visualization provided by software [72]. Thus, following this trend and the Web Utilization Miner solution [113], WELFIT represents client-side event logs and usage sessions in a summarized way trough directed graphs. However, due to the voluminous nature of client-side event logs, the graph representation must use some criteria of readability. WELFIT adopts the criteria mentioned in [72]: nodes should not intersect each other, intersection between nodes and unrelated edges, and edges should avoid crossing other edges.

Comparisons were performed to find an effective visualization of the usage graph containing the nodes representing triggered events. These comparisons considered graphs with and without cycles, and also labeling usage graph nodes using Web page element names, event names, and timestamp. In this graph visualization, the flow is the representation of transitions from one node to another.

If we consider timestamp to label nodes, the number of nodes of the resulting graph is huge, leading to a poor usability graph. Without considering the timestamp, many cycles arise and clearer graphs are generated. These cycles represent sequences of the same event at the same web page element, a common pattern of data found in client-side event logs

[102]. With this representation the maximum number of nodes is given by the product of the total Web page elements and the number of events tracked. Thus, the maximum number of nodes is not influenced by the number of tracked sessions, allowing the number of sessions to scale up, still resulting in a compact graph representation. Moreover, events related to the same target element were clustered to represent client-side usage flow in a consistent and synthetic way (Figure 2.1).

Thus, using walks performed by users, which can be defined as non-empty alternating sequence of vertices and edges [33], we used the procedure presented in Table 2.1 to build usage graph for a page. The procedure also shows how the approach used by the WELFIT data-logger deals with session tracking through the use of session identifiers and how the logged data is selected through data base reading operations. They are straightforward in comparison to session tracking and data cleaning procedures performed when using server-side logs as data source.

According to the rationale behind using cycles, layouts based on trees and graph drawing tools that support only trees are discarded. Moreover, graphs returned by circular layouts engines contained a number of crossing edges, resulting in graphs with poor readability. Finally, the software used to generate the clustered graph was Graphviz [1]. To represent the graph structure we used the JGraphT [61]. Once defined the graph's structure and elements, the challenge resides in configuring the graph drawing tool so that it can successfully generate graphs in a response time that keeps administrator's attention when generating reports.

2.5 Usage Patterns

With the structure and data obtained from the usage graph, it is possible to extract information of walks with greater edge weight values or to check the most performed transitions between nodes. In dense graphs, simple filters for deleting edges based on weight or transition percentage can reduce the overall graph representation and improve readability, keeping only the most repeated patterns that show the main sequences of triggered events when users interacted with evaluated Web pages.

Once having the usage graph structure, it is necessary to evaluate the walks present in the graph in order to find differences between walks to analyze accessibility and/or usability problems. Thus, to highlight flows in order to represent differences present in the event sequences we used the Sequence Alignment Method (SAM) approach presented in [49] and defined a heuristic to identify possible accessibility barriers and/or usability problems based on SAM properties.

SAM, also called Edit Distance, is a distance-based technique that represents the amount of operations necessary to equalize sequences [49]. Then, the lesser the distance

```
input  : Website and page's URL
output: Usage graph structure
read sessions of website and URL;
g ← new graph;
g.addVertex( start );
g.addVertex( end );
foreach session in sessions set do
    read events for session ordered by timestamp;
    previousVertex ← start;
    distance ← 0;
    events ← start + events + end;
    foreach e in events set do
        v ← new vertex;
        if e.targetId != null then
            v.name ← e.targetTag + "," + e.targetId;
        else
            if e.targetName != null then
                v.name ← e.targetTag + "," + e.targetName;
            else
                v.name ← e.targetTag;
            end
        end
        v.name ← v.name + "," + e.type;
        if !g.containsVertex( v ) then
            v.n ← 1;
            v.du ← distance;
            g.addVertex( v );
        else
            v ← g.getVertex( v.name );
            v.du ← ( v.n * v.du + distance )/( ++v.n );
            g.setVertex( v.name, v );
            if g.containsEdge( previousVertex, v ) then
                edge ← graph.getEdge( previousVertex, v );
                g.setEdgeWeight( edge, edge.weight + 1 );
            else
                g.addEdge( previousVertex, v );
            end
            previousVertex ← v;
            distance++;
        end
    end
end
```

Tab. 2.1: Procedure used to build the usage graph between two sequences, the similar the sequences are. This method reflects structural information through the order of elements within sequences, contrarily to the commonly used distance measures based on Euclidean distance [49].

The defined heuristic aims at using the average distance to a certain node and how this measure changes across the flows in the outgoing nodes. It allows an approximate comparison of distances between more than two sequences at once and at every node with an out degree greater than one. The heuristic is based on a metric that aims to evaluate the distance between nodes (i.e., the number of events triggered between every event).

The intent of the heuristic is to point out transitions of the usage graph that may represent design issues, using the fact that different length sequences require insertion/deletion operations when using SAM. Next we present some heuristic's properties. First, be G the digraph representing page elements of a certain Web page and a certain node of G, say u, has a average distance d_u. Additionally, be u's neighbors $v_0, v_1, ..., v_k$, and $d_v0, d_v1, ..., d_{vk}$ the respective average distances. Thus, if $d_u > d_{vi}$, then the in degree of v_i is greater than 1, meaning that it exists some alternative walk to v_i that does not have u as precedent node and has a smaller average distance. Thus, the walk from the root to u is not the short one, evidencing a performance difference and indicating that u is part of a walk that may contain an accessibility barrier and/or usability problem.

Finally, to visually represent these performance differences a minimum and a maximum threshold were applied. They are used to define colors for each node according to the following:

- RED) If the average distance value is greater than the maximum threshold, then the node needs to be marked as a part of a potential barrier/problem;

- YELLOW) If the average distance value is greater than or equal the minimum threshold and is lesser than or equal the maximum threshold, then the node does not reveal any novel information since it has a value in the expected interval;

- GREEN) If the average distance value is bellow the minimum threshold, then it represents a shortcut to other walks with greater average distances.

2.6 Preliminary Results

The real data set used in this work was captured by WELFIT during the interaction of users of a research group website. Part of the website's audience uses assistive technology. This website was chosen because WELFIT aims at identifying usage flows including the barriers faced by users of assistive technologies.

The data captured during 161 days resulted in 136 presentations of the invitation. 9 users refused to participate, one of them informed that is assistive technology user; 84 users accepted to participate, 10 of them answered that are assistive technologies users. These 84 users triggered 127 sessions. All sessions totaled 321,568 client-side events and 410 different URLs across the evaluated website. To exemplify the amount of data that can be represented in a graph, the average number of visited pages per session was 3.23 and the average number of events triggered in a page, for each session, was 784.312, resulting in usage graphs representing, in average, 2,533 events.

In the usage graphs, the flows are compared based on the average distances walked by users from the first triggered event to all events triggered at all page's elements. Thus, differences in distances walked by users to a certain element, combined with the presented heuristic, reveal significant performance differences, indicating design issues. Figure 2.1 (left) shows the behavior of links with inconsistent size definition and how this issue was found using WELFIT (right); the usage graph represents an assistive technology user trying to click on a non clickable element; at the bottom, the outlier node indicated by the heuristic.

From the visual reports representing usage patterns it is possible to rebuild the interaction of assistive technology users with UI. Additionally, the usage graph represents how users interact with Web page elements, making possible to study the effectiveness of some design decisions. One example of found patterns is that assistive technology users access the accessibility toolbar containing skip links, font-sizes, etc, as one of the first navigational elements.

2.7 Discussion

This work presented how to process logged data and automatically generate usage graphs summarizing statistic information present in the client-side event logs of a number of sessions. The WELFIT model uses a JavaScript client-side data-logger helping with the session tracking. Also, the database that keeps all logged data turns the task of data cleaning straightforward by using SQL (Structured Query Language).

The visual representation of the graph structure and the filter parameters presented in this work allow the visualization of the flow of events of usage sessions of a Web page, representing common transitions in the usage graph, thus revealing usage patterns and marking outlier nodes with the color red, which represent potential design problems.

Revisiting the main characteristics of the WELFIT approach and the existing approaches, we highlight: the technique presented focuses on facilitating the task of bringing users to the evaluation due to null reconfiguration required. The user have only to accept the invitation; the user can stop the capture at any time; the evaluator does not have to

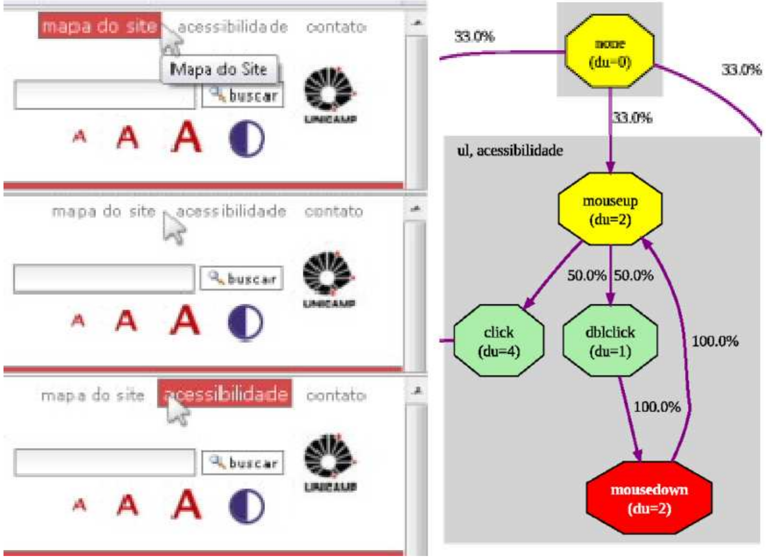


Figure 2.1: Mapping an UI design issue from usage graph extract (right) to the website's UI (left).

define task models, s/he just have do register the website. One drawback of WELFIT is that it requires knowledge about UI events and Web page elements to enable a specialist to understand the report. Thus, improvements suggested for the tool involves the use of icons to represent events and to incorporate an automatic generated text summary of the report.

Depending on the assistive technology being used, events can be triggered outside of the Web browser (e.g. screen readers). With this approach some screen readers may not trigger events at Web browser, thus hiding valuable information used by evaluation tools. This can be an obstacle, since in some cases data may not be captured by WELFIT; however, the approach used by screen readers is due to a limitation of HTML (HyperText Markup Language) versions earlier than version 5. The new versions of HTML allow all elements to gain focus, since *tabindex* attribute became a global attribute [130]. In addition, other assistive technologies were considered (e.g., screen magnifiers, widgets for font resizing, high contrast color scheme, etc).

The patterns presented by the tool represent a potential data source to incorporate personalization features. It is possible, for example, to avoid repetitive tasks like accessing the accessibility toolbar. This can be done through the use of agglomerative hierarchical clustering methods to identify clusters containing correlated event sequences and, consequently, similar ways to use the evaluated website. Once the clusters are defined, it is possible to support redesigns for each cluster found or to anticipate some events as the element that commonly receives focus. It is also possible to define global *tabindex* based on the most repeated patterns of navigation for each cluster of users, thus giving a global order of elements based on empirical data representing most common users' real behavior.

2.8 Acknowledgments

Thanks to PROESP/CAPES and FAPESP.

Chapter 3

Bringing Users of a Digital Divide Context to Website Evaluation Using WELFIT

Vagner Figueredo de Santana
Institute of Computing, University of Campinas (UNICAMP)
1251 Albert Einstein Av. Campinas, SP, Brazil
+55 19 3521-5838
vsantana@ic.unicamp.br

Maria Cecília Calani Baranauskas
Institute of Computing, University of Campinas (UNICAMP)
1251 Albert Einstein Av. Campinas, SP, Brazil
+55 19 3521-5870
cecilia@ic.unicamp.br

Abstract: The will of users to be part of a website evaluation depends on a number of variables (emotional state, context of use, etc). Especially remote evaluation tools must care about the invitation mechanism used. This work presents results of a participatory practice that observed how a diverse group of users deals with an invitation of a remote evaluation tool. WELFIT (Web Event Logger and Flow Identification Tool) is a remote evaluation tool that records client-side data during informal use of a website. This work aimed at verifying how the participants interact with the component of the tool added to a user interface of a well known application, the *Vila na Rede*, an Inclusive Social Network website. The results corroborate assumptions made during the requirements elicitation for the tool and also bring characteristics that would increase users willing to participate in this kind of evaluation. These characteristics would help evaluation tool developers to bring more users to remote studies involving client-side logging.

Keywords: Website evaluation, digital divide, accessibility.

3.1 Introduction

Automatic evaluation tools provide an important support for user interface (UI) designers and developers. They support maintainers of websites in many different tasks, such as validating code, applying guidelines verification, analyzing website's structure, and capturing and analyzing Web usage data. In website development, UI evaluation gains a special responsibility, since the design, development, and maintenance are influenced by other variables. The rapid pace and tight deadlines involving development and maintenance of websites affect negatively the effectiveness and reliability of the whole processes, from the detection of failures to the evaluation of benefits and costs of the proposed changes [15].

An UI evaluation can be remote or non-remote. Remote evaluation means that the user does not need to go to some test environment or UI lab to participate in the evaluation. Differently, non-remote evaluation requires the user to be in a controlled evaluation environment. When evaluating UI, the participant can make informal use (i.e., the evaluation requires the completion of freely chosen tasks) or formal use (i.e., requiring the completion of specifically selected tasks) [59]. Remote evaluation and informal use are interesting characteristics, since they prevent the UI usage to be biased during the evaluation. Furthermore, some authors argue that tests in controlled environments are artificial and may influence the results [96].

The automation of tools for UI evaluation may involve capture (i.e., logging usage data), analysis (i.e., identification of problems), and critique (i.e., suggestion of improvements) [59]. In addition, it may also involve adjustment (i.e., automatic customization of the UI in order to eliminate identified problems) [100]. Events can be defined as effects resulting from user's or system's action. They may occur at client-side and at server-side and often the collection of these events is called, respectively, client-side logs and server-side logs. Besides, UI events are natural results of using window-based interfaces and their components [51]. Events triggered at client-side are more detailed data sources of how users use a website [37, 83] than server-side data that can lead to less accurate results [23]. Client-side event logs allow keeping track of all events triggered by users (e.g. mouse movements, mouse clicks, or pressed keys) or by the browser (e.g. image is loaded, form automatically receives focus, some error occurs) [100]. On the one hand, server-side logs allow keeping track of users' flow in the website. On the other hand, client-side logs allow keeping track of users' flow in a Web page and in the website.

Remote evaluation is a topic that has increasingly gained attention in the Human-Computer Interaction (HCI) field. Current literature counts on works that exploit the value of real work context involving a broad range of users and tasks [48], expressing the validity of critical incidents reported by users [19], the eminent situation in which evaluators will turn remote computers into remote test sites [44], and the combination of questionnaires data with server-side logs [128]. Nevertheless these studies do not consider digital exclusion, accessibility, nor how to deal with sessions when the evaluations scale up, since some of the works focus on the synchronous remote evaluations, requiring the evaluator to be connected to the participant, in each session.

Online Social Networks are no longer a trend, they established in an irreversible way [108]. The communication mediated by computers widens the reach of social networks [18]. However, the dissemination of this kind of information system will only be possible if one considers the diversity of skills and competences of the user population [108].

Taking in combination evaluation tools and Online Social Networks, the focus of this work is to present how an evaluation tool can be considered in informal use, since it allows data capture when users are interacting with the evaluated UI in their usual (real) environments. In addition, this paper shows how users in a digital divide context interact with an evaluation tool during a participatory practice using an Inclusive Social Network. The main concerns of the study are related to data-logging, privacy issues, and users will to be part of an UI evaluation process. The tool used was the WELFIT (Web Event Logger and Flow Identification Tool) that uses both automatic capture and analysis, using client-side logs as data source [108]. It summarizes and visually represents the captured logs, allowing a HCI specialist to identify usage patterns and design issues as well.

One expects that a tool that successfully communicates with novice and digitally illiterate users would help HCI specialists to bring users to be part of UI evaluations, making possible the identification of accessibility barriers and/or usability problems. Another concern in this study is to improve both accessibility and usability (A&U) of the tool's invitation mechanism, since even if the services are accessible and the users can actually perform some tasks, it is also important that they perform tasks easily, effectively and efficiently [4]. The website used in this study is the *Vila na Rede*, an Inclusive Social Network, initially designed with/to the community of residents of *Vila União*, neighborhood on the outskirts of Campinas, *São Paulo*, Brazil. This website is a beta system being developed by the research group called *e-Cidadania*, which aims at studying and developing solutions to the interaction design and UI challenges related to information systems allied to the citizenship context. Inclusive Social Network is the social network where every person can integrate a group which is able to interact under a social protocol and a set of rules to promote the sharing and mobilization within the group [50]. The inclusive character can be attributed to a system that conforms interaction requirements favoring a wide audience and preventing a digital exclusion [4].

The next section presents the motivation and context of this work, and then tools and methodology used in this study are detailed. Afterwards, the results obtained are summarized. Finally, the last sections discuss and present conclusions obtained so far.

3.2 Motivation and Context

The Brazilian Institute of Geography and Statistics (IBGE) states that 14.5% of Brazilian people have some kind of disability. Considering the population older than 64 years this percentage grows to 54% [53]. Accessible interfaces are indispensable for personal autonomy and social inclusion. This is reinforced, in several countries, by special legislation protecting people from digital exclusion [4].

The wide diversity of physical, sensory and cognitive characteristics makes the UI design very complex [4]. Also, different competencies of users add complexity to the tasks performed by information system designers and developers, since 75% of the Brazilians are not fully literate, i.e., can not understand long texts neither relate their parts, nor prepare synthesis [57]. When barriers related to some kind of disability or literacy are not determinant, the barrier related to the lack of proficiency in using Communication and Information Technologies (CITs) arises, since it is considered the main barrier to the use of Internet in Brazil [20].

In such a scenario, unfortunately not restricted to Brazil, one can verify that even websites designed and developed considering A&U guidelines, may be involved in challenging contexts of use. These challenges involve the use of information systems by persons with multiple disabilities or considering several configurations of hardware and software.

Due to the great diversity of users' characteristics and context of use, it is almost impossible to consider all of them in the design phase [4]. In addition, these contexts of use are hardly covered during evaluations in controlled environments. Thus, support to continuous evaluation considering observational data represents a way of coping with these possible contexts of use in evaluations or redesigns contexts.

The *Vila na Rede* design process has counted on Inclusive Participatory Practices of the *e-Cidadania* project. This work details a study conducted during the 8th participatory practice of the project. These practices count on potential users representing the Brazilian scenario (Figure 3.1).

Users that participate in the study represent a subset of the diversity present in the Brazilian population. The set is called *Cenário**, or star scenario, as a metaphor to the transitive closure of Set Theory. This scenario aims at representing several users' abilities when using CITs. In the context of *e-Cidadania* project, 12 people from a local community, considering eventual absences, usually participate in the practices. Therefore, the *Cenário** represents a "snapshot" of the diversity, without having the pretension of exhausting possibilities or being complete [10].

Figure 3.1: Researchers and members of community working together during an Inclusive Participatory Practice.

3.3 Tools and Methodology

This work refers to a participatory practice in which 11 participants worked in two different activities to investigate:

1. The search engine of *Vila na Rede*, especially how users in this context deal with the semantics concerning searched terms;

2. The use of a prototype of a game with purpose, not used yet in the Inclusive Social Network.

The activity using the search engine was more relevant to study the interaction of users with the WELFIT in a context of a digital divide and was the one evaluated by the tool.

WELFIT is an evaluation tool that performs automatic capture of client-side events. It uses the resulting logs to point out possible UI design problems through the identification of usage flow [100]. The tool follows requirements for website evaluation tools based on event logs (Table 3.1).

These requirements were developed from the identification of solutions, limitations, and gaps of the evaluation tools studied. Considering the diversity of users profiles, these requirements will also be revisited in this work to verify the changes necessary to the

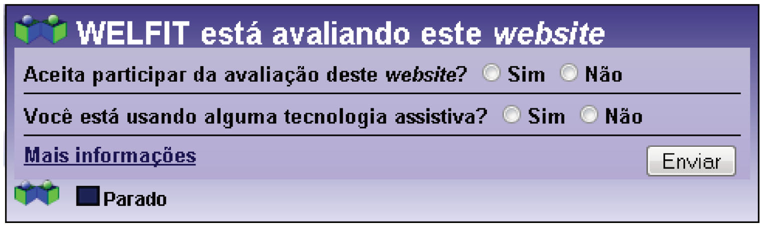
Semiotic Framework Layer	**Requirements defined in [103]**
Social	– Focus on the integration of A&U for the target audience of the evaluated website. – Enable remote testing during real use of the evaluated website. – Interfere with the Web page as minimum as possible.
Pragmatic	– Provide controls representing the status of the tool and user context during the test session. – The tool should use two actions: one to start the capture, which stays valid for future sessions, another to interrupt the capture, which may occur at any time. – Provide high levels of abstraction without depending on specific task models, grammars, or events.
Syntactic	– Use all available data (e.g., client-side events and server-side logs) in order to obtain correlations between them. The combination of the available data in different components can reveal information impossible to obtain independently.
Empirical	– Prevent that logs processing or transmition interfere with the use of evaluated interface. – The tool should implement safe and effective techniques without impacting on the website usage.
Physical	– Do not depend on resources or specific configuration of the participants devices (e.g., disk space, bandwidth, etc). – The evaluation tool should include mechanisms to achieve their goals in different configurations of hardware and software.

Table 3.1: Requirements for website evaluation tools based on event logs

requirements previously stated, now considering the real use in a context of digital divide and paying attention to the role of the evaluator, responsible for analyzing the tool's resulting reports.

The evaluation tool depends on the JavaScript support in the users' computer. JavaScript is a script programming language native in the newest Web browsers and is the technology used in the client-side module responsible for the data capture module of WELFIT.

The data capture starts as soon as the user accepts the invitation made by WELFIT, through an invitation widget (Figure 3.2). In conjunction with the question asking if the user wants to participate in the evaluation, s/he is also asked whether s/he is using any assistive technology. This information is used by WELFIT to build clusters according to usage patterns and to analyze problems faced by users in comparison with non-assistive technology users. In addition, obeying the requirements, once the user accepts the invitation, s/he has the power to stop the logging at any time s/he wants through the same widget.



WELFIT está avaliando este *website*

Aceita participar da avaliação deste *website*? ○ Sim ○ Não

Você está usando alguma tecnologia assistiva? ○ Sim ○ Não

Figure 3.2: Screenshot of WELFIT's widget.

WELFIT was chosen as the evaluation tool because the tool supports evaluation in the situation of informal and remote use. These characteristics allowed the addition of the tool into the participatory practice in a scenario closer to the real one; it was integrated to the real use of the *Vila na Rede*, and the practice took place at the community's telecenter, a public place where people have free access to CITs, training in basic informatics, etc.

The dynamics of the 8th participatory practice had the following steps:

1. A 15 minutes introduction presenting the 2 main activities (i.e., the search engine and the game with a purpose practices).

2. The search activity took place during the next hour in the IT lab of the telecenter, where usually the practices occur. The search activity task was executed in pairs, freely formed by the participants. During the tasks performed by users, the WELFIT was used without previous warning so that the observers could report on how users respond to the tool's intervention.

3. A 15 minutes coffee break discussing the activities performed, now including a discussion about WELFIT.

4. Finally, the last hour of the practice counted on the activity of the game with purpose, also with a presentation and a discussion. Since this activity is not the focus of this work, we will not detail this activity in this paper; more information on it can be found at [94].

During the whole participatory practice, the dialogs were recorded in audio. Each station counted on an audio recorder during the practice. Each computer was shared by a pair of users and each computer was accompanied by a researcher playing the role of observer. Also, a general video recording was made during the whole practice. To

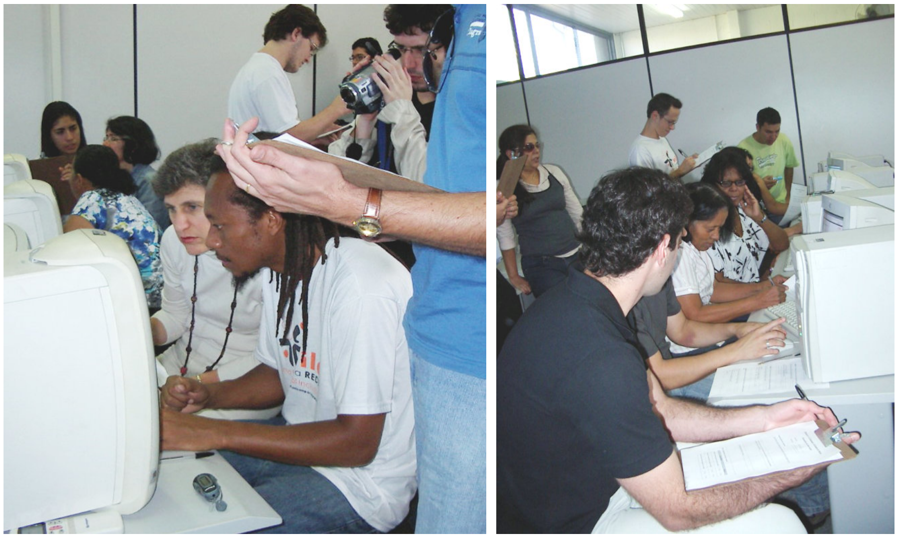


Figure 3.3: Part of participants and observers in the telecenter's IT lab during the practice involving the use of *Vila na Rede's* search with the WELFIT evaluation tool in background.

get an overview of the project, the reader can check the timeline showing extracts of all the practices occurred up to now at *e-Cidadania's* website. Regarding authorization of using the images of users, all participants signed a consent form allowing the audio/video recording and the use of their images as well.

Figure 3.3 shows pairs of users in each station, observers and their clipboards holding the observer forms. In addition, it also shows how audio and video recording were performed during the activity in which WELFIT was used.

The WELFIT study occurred in parallel with the search activity. Thus, the observer form used was written in a manner that would require little effort from the observers, since they were also observing the interaction of users with the search UI elements. The objective of the form was to verify whether the capture model used by WELFIT interferes with the way users of *Cenário** interact with the website being evaluated. In short, one aimed at verifying whether the data capture does not interfere with the website being evaluated and if the user understands the information presented in the tool's widget, since they were not aware of the evaluation and all required information should be present in the UI.

The observer form used contains 9 yes/no questions so that observers could answer the questions without interfering with the other activity. The questions included scenarios that could occur before the capture start and also the user's impressions of the tool (in the perception of the observer), information not captured by the tool's data-logger. At the end of the practice, observers were allowed to insert more detailed information and considerations to indicate user's feelings, since the Simplified Thinking Aloud Protocol could enable it. The Simplified Thinking Aloud Protocol is the technique of asking real users to think out loud while they perform the tasks [74].

The questions in the observer form were the following:

1. Did the user understand the message shown in the invitation?
2. Did the user show to be bothered by WELFIT's widget? Did s/he try to remove or close it?
3. Did the user have any doubt in relation to the terms used in the WELFIT's widget (e.g., recording, stopped)?
4. Did users answer the invitation?
5. Did the user accept to participate in the evaluation?
6. Did the user answer if s/he using any assistive technology?
7. Did the user access the link "more information"?
8. Did the user stop the event capture at any time?
9. If the user stopped the capture, did s/he reactivate the recording through the proper button?

The next section presents quotes and information gathered by the observers, audio recording, information obtained through WELFIT and also a revision of requirements for website evaluation tools based on logs defined in [103].

3.4 Results

Considering the filled observer forms and the audio recording of the interaction, this section highlights some obtained results and quotes. Table 3.2 presents the summary of the answers of observer's forms. Afterwards, we present significant dialog extracts considering the interaction of users with the tool's widget.

Approximately 90% of the users understood the message shown in the invitation. Around 81% of users felt bothered by the tool's widget. This fact can be related to the design solution, in which the widget is presented as an additional layer to the Web page being evaluated, in a fixed position defined by the tool.

In one station the widget was confused with an advertisement and the user just tried to close the invitation, answering only the first question.

In another station occurred the following dialog:

Station	**1**	**2**	**3**	**4**	**5**	**6**	**Total(%)**	**Total(%)**
QxA	**Y/N**	**Y/N**	**Y/N**	**Y/N**	**Y/N**	**Y/N**	**Y**	**N**
1	2N	N	2Y	2Y	2Y	2Y	90.91	9.09
2	2Y	Y	2Y	2Y	2Y	2N	81.82	18.18
3	2Y	N	2N	2Y	2N	2N	36.36	63.64
4	2Y	Y	2Y	2Y	2Y	2Y	100	0
5	2Y	N	2N	2Y	2N	2Y	54.55	45.45
6	2Y	N	2N	2Y	2N	2N	36.36	63.64
7	2N	N	2N	2N	2N	2N	0	100
8	2N	N	2N	2N	2N	2N	0	100
9	-	-	-	-	-	-	-	-

Table 3.2: Summary of answers obtained from the observer form.

– *"What is this thing? WELFIT is evaluating this website. Do you accept to participate in the evaluation? No, no, I don't want it now... just kidding*"

Then the users did not answer the invitation and skipped the invitation. But in the next page the invitation was shown again, and then the same user complained:

– *"How annoying! [...] Ok, I'll answer yes. Are you using any assistive technology? What is that? Oh, no! I don't want it anymore.*"

In this station the user tried to answer the first question, and the request to answer the assistive technology question caused him to refuse. The user did not know the term 'assistive technology' and then gave up in participating.

Roughly 36% of participants where in doubt in relation to the terms used in the tool's widget. Another transcription which shows this kind of hesitation is the one mentioned:

– (user) *What is it?*

– (observer) *It is an evaluation tool that the website is using.*

– *Do you want to participate in the evaluation? Oh, let's participate, then. What is assistive?*

– *Assistive technology is, for example, when a person does not see, and then there is a software to help him/her to navigate; a person who has lower vision has another software to support him/her... Thus, that is an assistive technology. For example, this thing to amplify the font size is also an assistive technology. Thus, in this case, are you guys using it? Yes or no?*

– *No.*

– *Then, no.*

– *But, we should use, shouldn't we? Because the letters are so small! (laughs)*

In another station users also showed to be somewhat bothered by the invitation and were in doubt whether they were or were not using an assistive technology.

– (user) *Someone has to take this off, right? Do you want to participate in the evaluation of this website? Yes, right? Are you using any assistive technology? What is assistive technical (sic)?*

– (observer) *Assistive technology would be a screen reader or any system that can help you to change this content, to help you to interact with the website.*

– *No, right? Thus, is it yes or no? No, right?*

– *It can be what you think it is right.*

– *Yes or no?*

This extract shows that the unknown term may misguide and distract the novice user. The user deduced that the invitation was related to the practice. Initially, the user did not want to participate, but just after asking the observer what is assistive technology s/he answered yes to the invitation. Afterward, participants did not understand where they would be carried after answering the invitation.

All users answered the invitation. 54% of users accepted to be part of the website evaluation and 36% answered that use assistive technology. After answering the invitation, during the search engine activity, users ignored the presence of the reduced widget, and also did not notice the event logging or the asynchronous transfer of logged data to the WELFIT's server.

After completing the search activity, the coffee break and discussion took place. The discussion counted with the presentation of questions concerning the search activity and, finally, commenting about the WELFIT's activity. Detailed information and results of the search activity can be found in [91]. Users felt comfortable in expressing their feelings and presenting suggestions to improve the interaction of the tool's widget with them.

From that discussion, the following interesting recommendations emerged:

- Allow the tool's widget to be included in the evaluated Web page as a layer or as a button;

- Use images to explain and give examples of what is assistive technology;
- Allow the customization of the tool's widget so that it can be gracefully combined with the evaluated Web page;
- Allow the insertion of the logo of evaluated website into the widget of the evaluation tool in order to gain users confidence;
- Allow the widget to be hided;

During the discussion, one user complained about the questions to be answered in the invitation and argued "Why the tool has to be so democratic?". This point is interesting since the user should have control over the capture, according to requirements presented in [103]. On other hand, they were surprised when noticed that some evaluation tools capture data without asking permission. In addition, they also suggested that this kind of tasks should be as short as possible.

Considering the results obtained so far and the requirements already stated (Table 3.1), one proposes a refinement of requirements for website evaluation tools based on event logs presented in [103], now considering a context of digital divide (Table 3.3).

The rationales behind the revision of the requirements for evaluation tools are the following:

a) According to the users' opinions and concerns, the widget communicating with them should show itself as part of the application UI to gain users' confidence. In order to do that the widget should consider customization and result in an integrated user experience. This leads to the insertion of 3 requirements;

b) Due to the amount of data gathered by an evaluation tool based in logs, the way that these data is presented to evaluators should be simple and represent user actions. In this direction one aimed at the 10th Maeda's Law of Simplicity, which states that "Simplicity is about subtracting the obvious, and adding the meaningful" [64]. This law was considered in order to focus in showing meaningful results of the tool. This leads to the creation of 3 requirements;

c) The requirement considering controls of the tool was moved from pragmatic to semantic layer since it is related to the current status of the tool and how it is presented to the user;

d) In this study, more than 80% of the users complained or showed to be bothered by the tool's widget. Thus, taking into account that the position plays an important role when communicating with users, a customizable position may avoid interfering with the evaluated website UI components;

e) Considering diverse contexts of use such as the use of computers in public places where personal information (e.g., cookie) can be not allowed, the tool should take this limitation into account and, for example, work together with the website being evaluated in order to record and retrieve users' answers and preferences as a profile feature. This leads to the creation of 1 requirement;

f) Due to the diversity context presented, the appropriated guidance should be offered if any specific term appears. This leads to the creation of 1 requirement;

g) Evaluation tools and Web Usage Mining algorithms studies commonly use server side logs [105], but these low granularity data does not help specialists in obtaining detailed information of how was the interaction of users with UI elements. Thus, considering high granularity, client-side data provides means of inferring low and high level interactions. This leads to the creation of 1 requirement;

h) Due to the amount of data obtained in UI evaluations and, consequently, the potential computational cost involved in presenting these data to evaluators, the time required to generate the report must consider practical limits and appropriate feedback.

The data obtained was input to guide the redesign of WELFIT's invitation widget (Figure 3.4). The redesign focused in presenting easy exits for users, it counts on a simple explanation about assistive technology and tries to cope with diversity in order to reach a wider group of users in website evaluations that are remote, during informal use, with real users during real tasks. Through the use of an adequate semantic markup an input field should also be included so that evaluators can customize the widget's appearance by inserting their own Cascading Style Sheets.

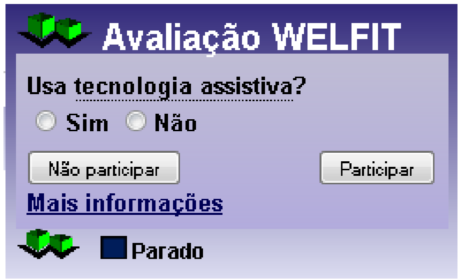


Figure 3.4: Screenshot of redesigned WELFIT's widget.

Although the WELFIT's reports were not the main focus of the work, it was possible to identify patterns related to the task considered in the search activity and the interaction with tool's widget. The reports highlighted events indicating repeated mouse movements over the search button before pressing it. This reveals the users' lack of fluency in using the mouse and/or that the button should be bigger for some users. It was also possible to identify situations in which some users tried to move the tool's widget. Finally, considering the types of events triggered and, consequently, how these participants interacted with the evaluated website, the users that informed that were using assistive technology triggered more mouse events than non-assistive technology users. In average, 93% of events triggered by assistive technology users were related to the mouse; for non-assistive technology users this percentage was 85%. Moreover, assistive technology users used the keyboard less than non-assistive technology users; 6% against 11% of triggered events. A hypothesis to that is that the maximum font size offered by the accessibility toolbar's amplifier (used by assistive technology users) should be bigger, since they required more mouse movements in order to point at elements and perform the same set of tasks of users that were not using the font size amplifier.

3.5 Discussion

From the results obtained it was possible to infer that key aspects should be considered for the particular considered context. The gracefully integration of the tool's widget into the evaluated UI showed to be a key attribute, since it focuses on providing an unified experience for users. In addition, another relevant point is that the task of answering the invitation should be as short as possible to support users who are willing to take part in the evaluation and to provide a quick exit for users who do not want to participate. In the participatory practice considered in this work 54% of users accepted to be part of the website evaluation. In a study not focused in the context of digital divide WELFIT had an acceptance rate around 90% when involving digitally fluent users [105].

The will of users to take part in a website evaluation depends on a number of variables. Thus, the revisited requirements may represent a way of avoiding that users get distracted or misguided by an evaluation tool intervention; also they should increase users willing to be part of a remote evaluation, especially considering contexts similar to the Brazilian one. In addition, the presented requirements may help evaluation tool developers to bring more users to studies involving client-side logging.

The fact that users were working in pairs helped in the Thinking Aloud Protocol, since they either were expressing their feelings or commenting/prospecting decisions. Consequently, users were more comfortable during the participatory practice and, in conjunction with the audio recording, it was possible to extract meaningful quotes showing how they reacted when the evaluation invitation occurred.

The percentages related to the devices involved in the generation of events revealed that some users have difficulties in using the mouse and pointing certain elements. This can be helpful when considering automatic classification in data mining studies aiming at tailoring websites to users' needs since blind people that use screen reader do not use the mouse and that people that informed that were using the font size amplifier triggered more mouse events than people that do not used the *Vila na Rede's* accessibility toolbar.

3.6 Conclusion

This work aimed at studying how a diverse group of users as we found in populations, particularly the Brazilian, deals with a website evaluation tool widget and functioning. In addition, the requirements for evaluation tools were revised considering the context of digital divide.

To gather information about usage a participatory practice was conducted using the WELFIT evaluation tool and the requirements for evaluation tools were revisited based on the results of such practice.

Results revealed that most of users of *Cenário** understood and answered the evaluation invitation, and showed to be bothered by the widget. Moreover, about a third of participants was in doubt about terms used and answered that was using assistive technology.

Requirements for website evaluation tools were reviewed and justified changes were proposed considering difficulties faced by users and observations made during the participatory practice. In addition, the role of the evaluator was specifically pointed out in this revision of requirements.

It was possible to verify that the will of users to be part of evaluations is also influenced by the confidence in the tool widget considering its similarities with the website being evaluated.

Besides the widget issues, the capture model used by WELFIT do not interfere with the way users of *Cenário** interact with the website being evaluated. The model performs tasks as logging, compacting, and transmitting the logged data.

Challenging contexts of use emerge when considering digitally illiterate users and a proposal of continuous evaluation considering observational data represents a way of coping with these possible contexts. Thus, bringing more real users to website evaluations is a way of gathering more representative data and, consequently, helping the UI to be more inclusive.

With the continuous evaluation supported by WELFIT it is possible to identify usage patterns of the Web mediated by assistive technology, during real use, remotely, and allowing the number of sessions to scales up. The usage patterns present in the tool's reports showed sequences of actions that indicated real usage issues, providing useful reports to the evaluators with minimum configuration effort.

Further work involves dealing with the problems found and also submitting WELFIT to the revisited requirements so that it can also reach a high acceptance rate of users of a digital divide context, to the same extent that the tool has regarding digitally fluent users.

3.7 Acknowledgments

We thank all observers that were indispensable in this study. We also like to thank InterHAD and *e-Cidadania* research groups, and FAPESP/Microsoft Research Institute for supporting the *Vila na Rede's* project and also FAPESP for supporting this Ph.D. research (grant #2009/10186-9). Finally, we thank all referees for the helpful comments.

	Revisited requirements
Social	– Focus on the integration of A&U for the target audience of the evaluated website. – Enable remote testing during real use of the evaluated website. – Interfere with the Web page operation as minimum as possible. – The tool's widget should gain user's confidence so that s/he can participate without privacy/security concern (see rationale a). – The evaluation report should call administrator's attention to points where users may have faced problems using the website (see rationale b). – The tool's widget should allow the customization of its appearance to be incorporated into the evaluated website's design (see rationale a).
Pragmatic	– The tool should use two actions: one to start the capture, which stays valid for future sessions, another to interrupt the capture, which may occur at any time. – Tool's controls must be gracefully combined with the design of the website being evaluated. The intention passed to the user by the website and controls should result in an integrated experience (see rationale a).
Semantic	– Provide high levels of abstraction without depending on specific task models, grammars, or events. – Provide controls representing the status of the tool and user context during the test session (see rationale c). – The positioning of the tool's controls must avoid interference in the website usage (see rationale d). – The evaluation report should allow administrators to view, in one spot, data concerning visits of one or more users (see rationale b). – If the user can not define settings in a certain environment, the answers and other data entered by the user should be recorded and kept by the website being evaluated (see rationale e). – Specific terms of the Web domain (e.g., assistive technologies, logs) should be avoided in a diversity context. If any appear, must have a text/image/audio explaining the term (see rationale f). – Elements used in the report should represent user actions (see rationale b).
Syntactic	– Use all available data (e.g., client-side events and server-side logs) in order to obtain correlations between them. The combination of the available data in different components can reveal information impossible to obtain independently. – Use high granularity data (e.g., client-side events) in order to allow the discovery of low and high level patterns (see rationale g).
Empirical	– Prevent that processing or transmitting logs interfere with the use of the evaluated interface. – The tool should implement safe and effective techniques to transfer logs without impacting on the website usage. – The time required to generate the report must consider practical limits if the intention is to retrieve it synchronously to the administrator. If is not the case, the tool should use an asynchronous method to inform the administrator (e.g., email, SMS) (see rationale h).
Physical	– Do not depend on resources or specific configuration of the participants devices (e.g., disk space, bandwidth, etc). – The evaluation tool should include mechanisms to achieve their goals in different configurations of hardware and software.

Table 3.3: Revisited requirements for website evaluation tools based on event logs; rationales of the review are pointed out.

Chapter 4

Visualizing User Interface Events: Event Stream Summarization Through Signs

Vagner Figueredo de Santana and
Maria Cecília Calani Baranauskas
Institute of Computing, University of Campinas (UNICAMP),
Albert Einstein Av., 1251, Campinas, Brazil
vsantana, cecilia@ic.unicamp.br

Keywords: Semiotics, User Interface Events, Visualization of Usage Data, Log Visualization, Event Icons, Icon Library, Image Library, User Interface Evaluation, Usability Evaluation, Accessibility Evaluation, Web Usage Mining.

Abstract: Effective visual representation is related to how people interpret signs created to carry specific information. In the last years many user interface evaluation tools are considering detailed usage data to represent users' actions. The volume of data gathered is leading developers to represent usage in a summarized way through graphical representations. If visual components used to represent complex data are not effective, then graphics used to summarize data may turn the interpretation of complex terms even harder. This work presents a study about graphical representations for user interface (UI) events and contributes with the validation of usage graph visualization and an open set of signs to support the summarization of client-side logs. The study involved 28 Information Technology specialists, potential users of UI evaluation tools. From the results one expects that evaluation tool developers, evaluators, and Web usage miners can reuse the validated usage graph representation and proposed set of signs to represent usage data in

a summarized way.

4.1 Introduction

The evaluation of user interface (UI) is a key task when developing information systems and is part of a number of Software Engineering development processes. UI evaluation represents a way of verifying whether the whole system is communicating effectively and efficiently with users. In the Web, the heterogeneity of UIs and the wide range of UI elements that designers can use when composing UIs reinforce the role of UI evaluation.

Website evaluation can be made remotely or non-remotely. Non-remote evaluation requires participants to move to some controlled environment (e.g., usability laboratory) while remote evaluation allows participant and evaluator to be separated in space and time, without requiring them to move to a controlled environment [59]. Thus, remote evaluation allows users to participate in an evaluation from anywhere, a key characteristic when evaluators want to consider accessibility or mobile devices.

Events can be defined as effects resulting from user's or system's action. They may occur at client-side or at server-side and often the collection of these events is called, respectively, client-side logs and server-side logs [104].

In the last decade, website evaluation tools using server-side data (i.e., based on Web server logs) became popular. They are used to analyze a number of metrics such as page-views, visited Web pages, referrers, landing pages, etc. Examples of tools that use server-side data are: Web Utilization Miner [113], WebSift (Web Site Information Filter) [25], WebQuilt [52], LumberJack [21], WebCANVAS [16], and DCW (*Descubridor de Conhecimento en la Web*) [35].

On the other hand, data capture at client-side allows evaluators to discover more precisely how a UI is used, since one page-view may be represented by a stream of hundred of events representing the user's behavior. This characteristic makes client-side data a more adequate source to represent details of the interaction of users with UIs. However, using this data source also brings challenges concerning logging, transferring, summarizing, and presenting logged event streams. Examples of tools that use client-side data are: WebRemUSINE (Web Remote User Interface Evaluator) [83], WAUTER (Web Automatic Usability Testing Environment) [9], MouseTrack [7], MultiModalWebRemUSINE [84], UsaProxy [8], WebInSitu [13], Google Analytics [43], WELFIT (Web Event Logger and Flow Identification Tool) [105], WebHint [122], and WUP (Web Usability Probe) [17].

Considering the presented evaluation tools, it is possible to verify that there is a trend in the last decade towards the use of client-side logs as data source. In addition, the summarization of the captured data appears as vital task in order to get the behavior data contained in hundreds of log lines.

The literature counts on works that deal with the issue of representing behavioral data. The visual representation commonly considered in these works is via graphs, which allows the visualization of patterns (through edges' attributes) and actions performed by users (through nodes' attributes) [105, 113]. In addition, Mutzel and Eades [72] reinforce that graphs are the most common form of visualization provided by software.

In the context of evaluation tools, evaluators should easily grasp users' behavior when analyzing tools' reports. Usage graph is a type of report containing a directed cyclic graph in which nodes represent events occurred in a Web page and edges represent the sequence in which events had occurred [104]. A usage graph representation was proposed in [105] after a comparison considering different representations of behavior through graphs. In the mentioned study authors presented that the maximum number of nodes is given by the product of the total Web page elements and the number of events tracked, not depending on the number of tracked sessions. The presented solution is a graph containing only textual data, which makes it difficult for an evaluator to analyze a usage graph representing thousands of events. In addition, such usage graphs require evaluators to know all events represented in the nodes, which usually is not the case as we will detail in Section 4.4.

Considering the previous mentioned works and trends as main motivators, our research aims at presenting such usage graphs in an efficient manner, converting as many textual information as possible into signs. Thus, the main goal of this work is to represent events through the use of icons. According to Peirce [86], icons are the only way of directly communicating an idea.

The Peirce's Semiotics counts on deep studies regarding signs. Moreover, Peirce presents rich taxonomies and different and efficient ways of classifying signs in a precise way. The thorough study of signs made by Peirce corroborates the use of his works as the main theoretical reference.

In this context, this work contributes with the validation of a usage graph representation and the proposal of a set of signs to represent UI events. The set is open and is available for the HCI (Human-Computer Interaction) community at `http://argos.nied.unicamp.br:8888/welfit/images/`. The set was designed, evaluated, and validated. These phases will be detailed in the following sections. Regarding the evaluation of the designed signs, works of Rubin [96] and Wainer [126] guided methodologically the experiment design, forms composition, bias avoidance, and conduction of evaluations.

This work is organized as follows: the next section summarizes the theoretical basis and the rational of the proposed signs; section 4.3 details the evaluation methodology; section 4.4 presents the results, and section 4.5 concludes and shows further directions.

4.2 Background

It is not difficult to find open icon libraries for developing websites or GUI (Graphical User Interface), but there is no such availability of open library to represent UI events, indicating the need of such set of icons. A popular example of icon library is the Open Icon Library [80]. It is a consolidated source of icons for people to customize UI. It offers a free resource for developers looking for icons to use in their free/open projects and has more than 10,000 icons; none of them refers to UI events.

This work is theoretically grounded on Peirce's Semiotics. Semiotics can be defined as the discipline that studies signs and systems of signs. A sign (or *representamen*) is something that, under certain aspect, represents something to somebody, i.e., creates – in the mind of a person – an equivalent or a more developed sign (interpretant). Sign represents an object, not obligatorily in all of its aspects, giving an idea of the represented object [86].

Peirce presents properties and details signs based on trichotomies. This work follows the most important trichotomy in which a sign can be classified as an icon, an index, or a symbol. The icon (Figure 4.1, A) is a sign that refers to the object as a result of *representamen*'s characteristics. From its observation it is possible to discover characteristics of the object being represented. For example, a house drawing presenting its main characteristics (i.e., walls, door, and roof) in simple lines refers to the proper house object. The index (Figure 4.1, B) is a sign that refers to the object that it denotes as if the *representamen* was directly affected by the Object. An index has the cause-effect relationship between object and *representamen* and can also be seen as an organic pair between the *representamen* and the object. For example, when seeing smoke coming from a chimney the smoke is the effect that makes you think about what caused it. The symbol (Figure 4.1, C) is a sign that refers to the object it denotes by virtue of an established convention, law, or rule. For example, a road sign presenting the letter 'P' may indicate, by an established convention, a parking lot [86, 92].

Considering the chosen data source, the signs proposed to represent UI events are based on standard events (Table 4.1).

Bearing in mind that the only way of directly communicating an idea is through an icon [86] and that reports displayed to evaluators should present the big picture of users' behavior [104], then the rationale of the design of the signs to represent UI events focused first in creating effective icons. Then, in case of signs failing to be represented as icons, the fall backs were index, and, lastly, symbol.

It is worth mentioning that events related to concrete actions of users that are at users' and evaluators' sight were easier to represent as icons (e.g., click). However, signs representing events triggered by the browser (e.g., load) or as direct consequence of events

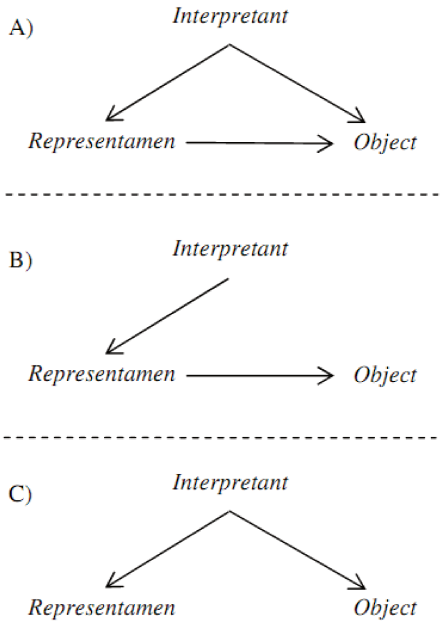


Figure 4.1: Relationship of terms of the trichotomy that defines: icon (A), index (B), and symbol (C).

triggered by users (e.g., change) were harder to represent as icons, falling back to symbolic or indexical representations. The relationship among these UI events and the classes of sign considered resulted in a mapping that supports the creation of new signs and it will be presented in the results section.

The creation of the signs involved a base element to represent a window-like abstract UI element, as presented in Figure 4.2.

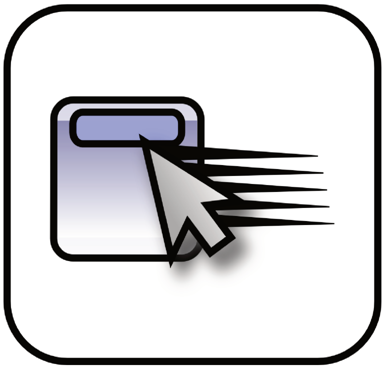


Figure 4.2: The Sign representing the *mouseover* event.

UI events are commonly related to movements just performed. Thus, in order to

Event	**Triggered when...**
Abort	the loading of a document or an image is cancelled
Blur	an element loses focus
Change	the content of a field changes
Click	the mouse clicks an object
Dblclick	the mouse double-clicks an object
Dragdrop	an element is dragged and dropped in a new position
Error	an error occurs when loading a document or an image
Focus	an element gets focus
Keydown	a keyboard key is pressed
Keypress	a keyboard key is pressed or held down
Keyup	a keyboard key is released
Load	a Web page or image is finished loading
Mousedown	a mouse button is pressed
Mousemove	the mouse is moved
Mouseout	the mouse is moved off an element
Mouseover	the mouse is moved over an element
Mouseup	a mouse button is released
Move	a window is moved
Resize	a window or frame is resized
Reset	all the content filled in a form is deleted
Select	a text is selected
Submit	a form is submitted
Unload	the user exits the Web page

Table 4.1: Standard UI events considered in the study [124]

represent them graphically, photographic streaking effect presented by McCloud [68] was added in order to represent movements, actions performed, and state change (Figure 4.2).

In order to build other signs, the base UI element was combined with elements inspired in well known UI components (e.g., pointer and hand) and personal computer hardware (e.g., mouse and keyboard keys). However, some events are not triggered directly by users, for instance, load and abort. This reinforces the need of evaluating signs in order to represent this kind of events to evaluators.

4.3 Experiment Design

The first set of signs was analyzed in an evaluation counting on 15 participants of a graduate discipline on Special Topics on HCI. The second set of redesigned signs counted on 13 participants of a graduate discipline on Design Patterns. Both of the classes were formed by software engineers that are potential users of such signs representing UI events.

There is no intersection or contact among these participants in order to avoid bias related to previous experience considering the interpretation of the evaluated signs, reports, and evaluation forms.

The second evaluation was done 9 months later, based on results of the first evaluation; this means that the signs were redesigned based on results of the first evaluation and then checked in the second evaluation. These two groups of participants were chosen because their profiles are part of the target population considered (i.e., potential users of UI evaluation tools). They are researchers, students, and professionals that would use an evaluation tool to analyze users' behavior.

The evaluations had three printed forms (A, B, and C) and a questionnaire to verify the representations used in the usage graph report. With these forms we also gathered data concerning gender, age, and profession of the participants. The instruments are detailed as follows.

Form A investigates the activity of interpretation of signs without context; this means that the signs were not presented in a meaningful order. The form has a 4 x 6 table containing the 23 proposed signs in random order, since some of them have a direct relationship (e.g., keydown-keypress) and placing them together or in alphabetical order might influence results. Along with each sign there was a bracket gap to be filled with an index representing the filling order and a gap to be filled with the meaning that the sign has for the user (e.g., the gaps pair [_] _____ could be filled as [1] click). Regarding instructions, the form A asked participants to write down the meaning of each image.

Form B focuses on presenting to participants a usage graph report representing a real usage of a Web page being evaluated by WELFIT [104], one of the studied tools that considers detailed data. In the form B the participants were asked to write down the meaning of the usage graph report representing the usage (Figure 4.3). In other words, they were asked to identify the meaning of signs in a situated context.

The usage graph report uses the proposed signs in logical and meaningful sequence (e.g., blur-focus, keydown-keypress-keyup, mousemove-click). The usage graph was designed to help the identification of the detailed interaction of users with UI elements. Regarding instructions, the form B asked participants to describe what might have happened during the usage represented in the usage graph. It is worth mentioning that Figure 4.3 was resized in order to present the whole usage graph, just as would occur when using an evaluation tool if zoomed out; in this case the textual information are almost unreadable, but the signs can be identified. This example presents another context that motivates this study.

Form C was given to participants only after finishing forms A and B. The form C was used as a matching exercise between the signs and their intended meanings, using the

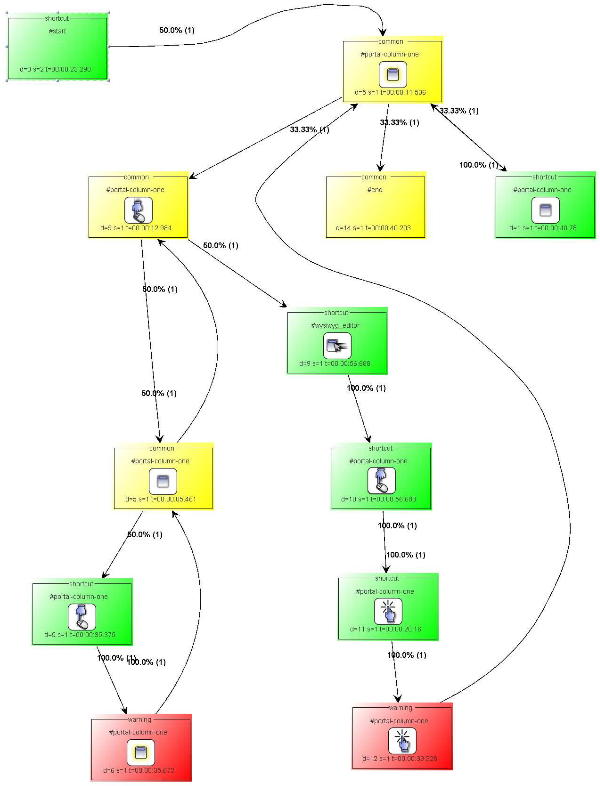


Figure 4.3: Overview of the usage graph that was part of the form B, representing the evaluated signs in a situated context.

indexes that participants had filled in the form A. This was done in order to verify the accuracy of the signs in a context of an Information System in which they will count on a legend to get signs actual meanings.

The final questionnaire was presented in order to try to identify weak points in the representation contained in the form B concerning information added to nodes.

The procedure of each of the two evaluations was the following:

1. At the first moment, half of the students (plus/minus one) received first the form A and then (10 minutes later) the form B. This group of students is referred from now on as group AB;

2. The other half received first the form B then the form A, referred from now on as group BA. This was done in order to verify the influence when participants were trying to identify signs' meaning without context (before the usage graph report containing the signs in a meaningful order) and vice versa;

3. Lastly, once both groups had filled up the forms that were given, then all participants

received the form C and the questionnaire.

4.4 Results and Discussion

The accuracy was measured considering the term filled by participants in the form A and if they matched the designer's pragmatics. If the term filled by respondents refers, in an unambiguous way, to the action/event being represented, then the sign was considered successful in communicating its meaning to the participant. For instance, one participant filled the term 'click' for the mouseup sign; then it was counted as not successful because there is another event named 'click'. Other participant filled the term with 'release mouse'; this was counted as successful. Table 4.2 presents the summary of evaluations and accuracy of signs. Considering participants' answers, the mean of answers that met the meaning of the event being represented, for each participant, were: in the 1st evaluation, 61.74% (standard deviation (s) of 19.11%); and in the 2nd evaluation 65.22% (s=15.68%). The low mean and high standard deviation of right answers per participant might be related to the following points: the strict and unambiguous analysis performed regarding the terms filled by participants, since some participants left blanks or filled the same term for more than one event; and, the difficulty of participants in defining events triggered by the browser.

Taking into account signs' accuracy, we obtained the following means: in the 1st evaluation, 62.61% (s=27.02%); and in the 2nd evaluation, 64.88% (s=25.28%). These results represented a small improvement considering redesigned signs.

The best results (accuracy > mean accuracy + s) were related to the signs representing the events: in the 1st evaluation, abort, mousemove, mousedown, and submit; and in the 2nd evaluation, abort, error, mousedown, and submit.

The worst results (accuracy < mean accuracy - s) were related to signs representing the events: in the 1st evaluation, change, click, dblclick, error, focus, and unload; and in the 2nd evaluation, change, mouseover, mouseout, and unload.

In the last case, unload and change events were also present, revealing the most difficult events to be represented, this difficulty on designing them will be discussed in the next section.

Regarding lack of responses, the first evaluation had 4 empty fields (in the 15 forms A), two of them referring to the change and unload events. In the second evaluation, the 13 forms A had 8 empty fields, two of them referring to dblclick event.

Regarding the order in which gaps were filled in form A, it is possible to check what signs had quicker interpretation from the users. The signs defined first by the users were related to the following events: in the 1st evaluation, abort, resize, dragdrop, and mousemove; and in the 2nd evaluation, abort, unload, dragdrop, and reset.

Attribute	**1st evaluation**	**2nd evaluation**
Participants	15 participants (12 males, 3 females)	13 participants (7 males, 6 females)
Mean age	28.35 years (s = 6.1 years)	28.09 years (s = 4.41 years)
Right definition for sign per participant (Total)	61.74% (s = 19.11%)	65.22% (s = 15.68%)
Mean accuracy of signs	62.61% (s = 27.02%)	64.88% (s = 25.28%)
Mean of correct matches between sign and event meaning (Total)	78.26% (s = 15.68%)	77.26% (s = 15.18%)
Mean of correct interpretations of the usage graph	40%	61.54%

Table 4.2: Summary of evaluations' results.

The last ones defined, indicating that their meanings were harder to grasp, were: in the 1st evaluation, mouseover, move, focus, and mouseup; and in the 2nd evaluation, dblclick, focus, mouseover, and mouseup.

Referring to the validation of the usage graph as summarized representation of event stream data (i.e., form B) an improvement was also obtained. In the 1st evaluation the usage graph was correctly interpreted by 6 out of 15 participants (3 from group AB and 3 from group BA). The main problem in the descriptions filled by participants was related to the click event, since 6 out of 9 participants that interpreted the usage graph differently from what was expected informed that the click event was something referred to an 'mark as favorite' action. This reinforces our rationale in combining the two types of evaluation presented in this work, i.e., the signs seen in isolation and within the usage graph. In the 2nd evaluation the usage graph was correctly interpreted by 8 out of 13 participants (4 from group AB and 4 from BA group). The main issue here was related to the fact that each usage graph node was thought as referring to a Web page, which usually occur in evaluation tools considering page-view as the navigational unit. Table 4.3 presents samples of redesigned signs that helped in improving these results.

Considering form C, which was used to mach the event meanings with signs of the sheet A, as a matching terms exercise, the successful matching had a mean of 78.26% (s=15.68%) per respondent; and in the second evaluation the result was 77.26% (s=19.40%). This reveals that if the system using these signs was using a legend, no significant improvement should be expected. According to this point and to the amount of information present in

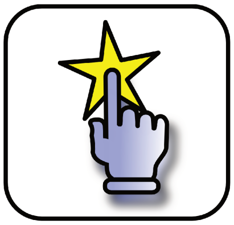
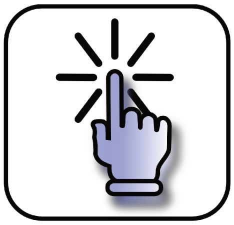
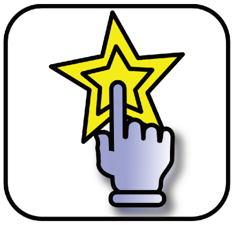
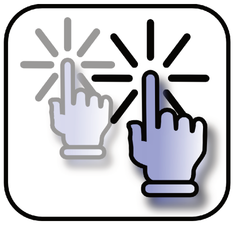
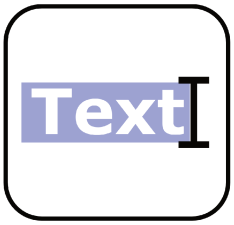
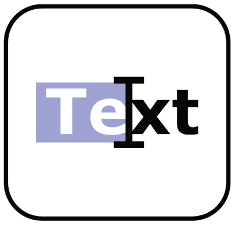
	1st evaluation	**2nd evaluation**
Event	Sign and accuracy	Sign and Accurracy
Click	33.33% 	61.54% 
Dblclick	33.33% 	46.15% 
Select	40.00%  	69.23% 

Table 4.3: Examples of redesign results.

a usage graph, it seems more adequate to consider tool tips than legend for the elements present in the usage graph. This suggestion was also made by some participants through the questionnaires.

The results obtained from the 1st and 2nd evaluations lead to some hypothesis considering the improvement of the signs in isolation and the usage graph. The hypothesis for the improvement in the accuracy of signs is that the redesign eliminated some of the elements that were leading to the misunderstanding on mapping signs to proper events, e.g., the click sign that was revoking the star element used in many websites for rating/ranking and the select sign that, after redesign, is representing more clearly the ongoing action. In addition, the hypothesis for the noteworthy improvement of the correct interpretation of usage graphs is that the redesigned signs improved the understanding of the whole graph and, consequently, the usage context. This point was reinforced by the fact that evaluators were not aware of all standard events of Web UIs, thus the interpretation of signs in a usage graph helps in decoding the signs considering the meaning of the whole context.

It was possible to check the differences regarding the evaluation of usage graphs and the interpretation for each single sign's meaning. Hence, the accuracy of signs is a key factor on understanding the entire usage graph. This outcome points out that, as presented before, interpreting the whole usage graph is easier than understanding the signs without context. However, it was also verified that improving single elements that compose the whole usage graph impacts significantly in grasping the meaning of the usage graph. In sum, the mean accuracy of signs improvement from 62.61% to 64.88% impacted on the improvement of the correct interpretation of the usage graph from 40.00% to 61.54%.

The difficulty of designing accurate signs was more present when referring to events that are distant from evaluators' perspective, i.e., is not part of the daily work of evaluators that do not work daily with Web pages event handlers. Consequently, it was harder to obtain a *representamen* to stand for such actions that, in turn, creates the desired interpretant in the mind of the participants. This was observed in different cases (e.g., unload and change events). In addition, after analyzing why some signs obtained better accuracy than others based on evaluations and on the Semiotics, we found a correlation considering the trichotomy and the categories of UI events. From that correlation, we present a mapping among the classes of signs and the three categories found (Table 4.4).

The three categories are related to events that are directly triggered by users, triggered as a result of events triggered by users, and events triggered by the browser as its natural functioning (i.e., without any direct connection with users events). The mapping can be used as a guide to design and organize new signs for representing client-side single events, composed events, and abstract events, since there are tools that consider this kind of client-side event abstractions, for example, Google Analytics [43] and WUP [17].

Candidate Class of Sign	**Event category**	**UI events**
Icon	Direct users actions	click, dblclick, keydown, keypress, keyup, mousedown, mousemove, mouseout, mouseover, and mouseup.
Index	Effect of users' actions or abstract events	change, dragdrop, move, resize, reset, select, and submit.
Symbol	Browser functioning	abort, blur, error, focus, load, and unload.

Table 4.4: Mapping relating events according to their sources and the candidate class of Sign to represent it.

4.5 Conclusion

Several user interface evaluation tools are collecting detailed usage data to represent users' actions. The volume of information demands a summarized way of presenting data through graphical representations. This paper presented a study on how to graphically represent detailed users' actions occurred at client-side, grounded on the Peirce's Semiotics. The proposed set of signs is a first approach to deal with the problem of the inexistence of an open library to represent UI events. The set of signs, now available to the Human-Computer Interface community at `http://argos.nied.unicamp.br:8888/welfit/images/`, was analyzed in order to adequately represent end users' behaviors to evaluators, achieving an accuracy that is close to the matching terms accuracy. In addition, the proposed signs were applied in a validation of usage graphs as a way of summarizing event stream data for evaluators.

A mapping of signs was presented, combining events, events categories, and candidate classes of signs to represent them. The mapping illustrates the complexity one has to deal with when designing icons in the context of usage visualization, especially when designing signs representing events that are not direct effects of users' actions. Thus, the mapping proposed may help designers who want to create signs for new UI events, guiding them in terms of what kind of sign to use and where to focus the pragmatics concerning the event to be represented.

The set of developed signs can be reused by other evaluation tools in order to represent users' behavior. Tools are gathering and presenting detailed usage data year after year, thus the HCI community is welcome to improve it.

Future works involve distributing the online versions of the forms and questionnaires used in this work to the community in order to allow the improvement of the proposed signs in large scale and to include new signs for events that are appearing along with emerging technologies (e.g., touch displays).

Finally, the complexity of UI is growing but events compose a defined set. Thus, in the very low level, UI events change a lot less than UIs, since they are coupled with technologies not with the use designers and developers make of it. New events are slowly appearing as those triggered by accelerometers. Even though, these new events can all be translated into signs and reported through usage graphs for analysis. Hence, a study regarding events of modern UIs and mobile applications are also considered for future work.

4.6 Acknowledgements

We thank all participants, colleagues that helped in the evaluations, and FAPESP - *Fundação de Amparo à Pesquisa do Estado de São Paulo* (grant #2009/10186-9) and CNPq through the EcoWeb Project (#560044/2010-0) for supporting this research.

Chapter 5

WELFIT: A Remote Evaluation Tool for Identifying Web Usage Patterns through Client-Side Logging

Vagner Figueredo de Santana,
University of Campinas (UNICAMP)

Maria Cecília Calani Baranauskas,
University of Campinas (UNICAMP)

Abstract: Although websites evaluation tools use different data sources (e.g., Web pages, server logs, and mouse tracks), few of them support remote evaluation using detailed observational data. Without considering data that represent the user's real interaction with the interface, usability problems and/or accessibility barriers may remain unknown. This work contributes to the field by providing a tool to identify usage patterns based on client-side event logs. The system records usage data during real use, identifies usage patterns, and indicates possible user interface design problems. The validation of the proposed tool involved a 15 month observational study of real usage of a website to verify how the tool is employed remotely during informal use. Results obtained are promising and point out that the tool enables examination on how assistive technology users use websites while performing real tasks remotely.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces – Evaluation/methodology. D.2.2 [Software Engineering]: Design Tools and Techniques – User interfaces

General Terms: Design, Human Factors

Additional Key Words and Phrases: Website evaluation tool, remote Usability evaluation, remote Accessibility evaluation, Web Accessibility, Web Usability, client-side events, user interface events.

5.1 Introduction

The Web is supporting more and more services from different knowledge domains, becoming essential for people's lives in our society. Some examples of resources and applications that promote this growth are communication support, entertainment, and government services. Nevertheless, analyzing Alexa.com's top 200 worldwide homepages that generate most user traffic, 96% of the sites fail to address known accessibility requirements from Web Accessibility Content Guidelines 2.0, such as to provide an adequate description of visual elements. In addition, considering this set of homepages, we also found that 97% of the sites have invalid HyperText Markup Language (HTML) code, and 93.5% have invalid Cascading Style Sheets (CSS) code. This reveals that there is much development work to be done in order to turn the Web into an accessible environment.

Accessibility and Usability (A&U) are playing an increasingly important role in the creation of a successful website [15]. In addition, Web applications contribute to enhance the social inclusion and autonomy of users with disabilities by enabling them access to education, labor, information, communication, leisure, etc. [4].

A&U differences and overlaps are constant topics of discussion in the Human-Computer Interaction (HCI) literature. In the Web context, there are two main referenced definitions for Accessibility and Usability, from the Web Accessibility Initiative (WAI) and from the International Standardization Organization (ISO). Web Accessibility means that people with disabilities can perceive, understand, navigate, interact with, and contribute to the Web [125]. Usability is the extent to which a product can be used by users to achieve goals effectively, efficiently, and satisfactorily in a specified context of use [58]. Moreover, an accessibility barrier is anything that makes it difficult or impossible for people with disabilities to use the Web [125]. Usability problems can be defined as user interface (UI) aspects that reduce the system's usability for end users [92].

If services are not accessible, they are useless for people with disabilities. Once accessibility has been achieved, it is also important that users can perform tasks easily, effectively and efficiently [4]. Thus, A&U are key requirements for information systems. Moreover, a lack of A&U integration may result in usable websites with low accessibility or accessible websites with low usability [27]. Bad designs complicate usage not only for people with disabilities, but for all users [4]. Thus, supporting Web Accessibility does not exclude users without disabilities and extends the concept of usability [103]. These characteristics reflect the overlap within A&U. Nevertheless, it is worth pointing out that Accessibility has been more closely related to the access and to the removal of barriers, while Usability has been more closely related to the attributes involving the actual usage.

Website evaluation is a way of knowing and removing accessibility barriers and usability problems to reach minimum A&U requirements. Due to the volume of data that commonly results from such evaluations, the use of automatic tools is necessary during the development cycles. Moreover, the usability of these tools is also essential, because they should not require great effort on the part of the evaluators when setting up evaluations or on the part of end users when participating in an evaluation. Automatic data collection guarantees the gathering of a vast amount of detailed data, which generally requires substantial effort and time in order to be properly interpreted by humans, in the absence of appropriate automatic data analysis techniques [85].

The total or partial use of automatic usability evaluation methods may reduce the time and costs involved in the development of Web applications because they liberate specialists from repetitive tasks such as manual log analysis [83]. Thus, in addition to the improvements that evaluation tools bring to both UI design and website development, they also promote consistency in the evaluations and allow the test sessions to scale up without increasing the evaluation costs.

When considering automatic evaluation tools, HCI practitioners count on two main groups of such tools: those that use Web page source code (content or structure) as data source and those that analyze usage data (logs). The current study focuses on usage data and will not detail evaluation tools based on Web page source code. Also, tools involving questionnaires (e.g., WEBUSE [22]) are not addressed in this paper.

The logs used by evaluation tools can be captured at either the server-side or the client-side. Capture at the server-side is technically simpler than capture at the client-side, but the data taken from the server-side only reveal information related to the page users have visited. In contrast, capturing data at the client-side is computationally more complex and may involve more tasks (e.g., capturing and transmitting logs), but such data reveal more detailed information (i.e., actions and UI elements where they occurred). Evaluation tools that use Web page source code or server-side logs are important components of UI evaluations. However, due to the heterogeneity of software and hardware, as well as of the possible contexts of use, evaluation tools that use client-side data may reveal details about the user interaction necessary to complement other evaluation results. Eckersley [36] argues that this heterogeneity of software and its plug-ins is meaningful to the point that they represent a device fingerprint, allowing the identification of devices with a high degree of certainty.

Evaluation results have direct impact on design rationales and some design decisions can be reconsidered based on them. This is directly related to the fact that the variety of needs and the wide diversity of physical, sensory and cognitive characteristics of users make the design of interfaces very complex. In addition, due to this diversity, it is almost impossible to consider all users in the design phase [4]. Although it is possible to identify different tasks and ways of using the projected UIs, questions related to different users' needs may appear in a number of contexts of use that were not foreseen before real usage. Citing ISO's definition, context of use involves the following variables: users, tasks, equipment (hardware, software, and materials), and the social and physical environments in which the product is used [58].

Logs captured at the client-side, which obtain details of interaction, are directly related to UI events since they contain information about the interaction of users through input and output devices. Thus, events are direct or indirect effects resulting from actions that can be originated either by the user or by the system. Examples of events that are the direct result of users' actions are: click and mousemove. Examples of events that are the direct result of system's actions are: load and error. Examples of events that are the indirect result of users' actions are: change and unload. Event stream is defined as a temporized sequence of events, which makes it possible to rebuild the actions performed during a logged interaction. Thus, a usage pattern can be defined as a set of combinations of event streams representing the behavior of one or more users. UI usage patterns show how users interact with UI elements and represent a source of information that opens new scenarios regarding how to evaluate UI design and how to adjust the UI as well.

Some authors advocate that logging techniques should be combined with data analysis features in order to provide useful results for the evaluators [85]. Thus, the proposed tool tries to combine the straightforwardness of logging data with the usefulness of summarized results to highlight potential usage incidents.

In this work we define usage incidents as a reference to the term critical incident, which was used in remote usability evaluations by Hartson et al. [48] and Castillo et al. [19]. Thus, critical incident reflects a usability problem, a missing functionality, or other ways in which a system fails to meet user needs. Hence, we define usage incident as a critical incident that also indicates an accessibility barrier.

The objective of this paper is to present a tool that supports the evaluation of websites so that they can be more accessible and usable by all. The proposed tool supports remote evaluations using client-side events. It focuses mainly on the identification of behaviors that suggest usage incidents including those coming from assistive technologies (ATs) users. The tool validation involved one case study that investigated the following hypothesis:

1. event streams triggered by AT users and non-AT users differ in the distribution of event types that compose them;

2. the accuracy of the heuristic used to point usage incidents is greater than 65%.

This paper is organized as follows: section 5.2 characterizes website automated evaluation and discusses literature related work to situate our proposal; section 5.3 presents the rationale and details regarding implementation aspects of the proposed tool; section 5.4 presents results of the proposal validation; and section 5.5 concludes.

5.2 Background and Related Work

Ivory and Hearst [59] proposed a taxonomy of techniques for usability evaluation automation and Hartson et al. [48] presented a comparative study involving remote and non-remote usability tests. These works show that website usability and accessibility automatic evaluation has several characteristics which involve: the participant's location, use of specific tasks, type of interaction between users and evaluators, data source, among others. The choice of each of these characteristics brings pros and cons to the evaluation process and results. In the following section we discuss the combination of different characteristics within the context of Web Accessibility.

5.2.1 Characterizing website automatic evaluation

Regarding the location of the participant, UI evaluation can be conducted remotely or non-remotely. Remote evaluation means that the user and the evaluator are separated in space and/or time; thus, the user is not required to go to a test environment or lab to participate in the evaluation. Conversely, non-remote evaluations require the user to be present in a controlled environment. Regarding the tasks involved in UI evaluation, the participant can make informal use of the system, when the evaluation requires the completion of freely chosen tasks, or formal use, when it requires the completion of tasks specifically selected for that goal [48, 59].

Considering location and use that participants make of the UI, we have gathered the following arguments. Tests in controlled environments are artificial and may influence the results [96]. It is important that users interact with the application being evaluated in their daily environments, but it is impractical to have evaluators directly observing users' interactions [84]. The remote work setting has become an intrinsic part of usage studies and it is difficult to have it reproduced in a laboratory setting. Moreover, developers often have limited access to representative users for usability testing in the laboratory [48]. Also, the real scenario of use plays an important role when dealing with accessibility due to difficulties in replicating the diversity seen in the configuration of specific hardware and software (e.g., ATs such as screen readers, magnifiers, etc.) used by the participants [13]. Thus, the tool proposed in this work combines characteristics of remote tests that occur during the informal use of a target UI, as a way to avoid biases in the UI use during the evaluation.

Regarding the relationship between users and evaluators, Paternò and Santoro [85] point out aspects in favor of the remote interaction. Remote observation allows the evaluator to observe the actual behavior in real time. Remote questionnaires allow users to provide feedback through a series of questions made available electronically. Critical incidents reported by the user allow users to directly report critical incidents to the evaluator when an incident occurs. Automatic data collection allows the compilation of different types of data regarding user behavior.

Considering the characteristics of those types of interaction, we can make the following considerations. Remote observation depends on the bandwidth of the connection between the user and the evaluator. Remote questionnaires depend on the users' willingness to answer the questionnaire. Critical incidents depend on the user's knowledge in identifying critical incidents. Although automatic data collection has limitations when considering subjective data and the user's behavior in real time, it allows the number of test sessions to scale.

Tools for the automatic evaluation of UI may involve support for data capture (i.e., logging of usage data), analysis (i.e., identification of problems), and critique (i.e., suggestions for improvements) [59]. They may also involve UI adjustments [100]. Considering data capture, there are three main methods used to gather website usage data: capture at the client-side, capture at the server-side, or proxy-based capture. Each of these methods is discussed next.

Capture at the server-side may occur through the use of the Web server access log. This kind of data is the main data source for Web Usage Mining (WUM) tools [88]. WUM is "the process of discovering and interpreting patterns of user access to the Web information systems by mining the data collected from user interactions with the system" [111]. WUM was proposed as an area where Data Mining methods are unified and applied to Web data [88]. Data Mining can be understood as the analysis of large data sets to detect non-trivial relationships and to summarize these relationships in a useful and easy to understand way [46, 120]. Examples of WUM tools were proposed by Spiliopoulou and Faulstich [113], Cooley et al. [25], Chi et al. [21], and Domenech and Lorenzo [35].

Server-side logs have several shortcomings. Their effectiveness is strongly limited by the impossibility of capturing local user interactions and by the validity of the server logs that cannot capture page accesses stored in the browser's cache. If a user clicks on the browser's back button and the content retrieved is a cached Web page, then this will not reach the server and, thus, will not be logged. In addition, interpreting the actions of an individual user is extremely difficult because the methods for capturing and generating server-side logs are not designed for gathering useful usability data [38, 85]. The strength of this approach is the low cost of obtaining the data because they are a natural product of the Web server functioning. However, the identification of the users' tracks and sessions is more complex than when using client-side or proxy-based approaches. In addition, server-side logs do not contain detailed information about users' actions during interaction. Instead, this kind of log contains only Web pages visited by the users (page-views).

Proxy-based capture is an approach in which the tool or data-logger mediates between the user and the websites that s/he accesses. Thus, when the user accesses a Web page, the request is sent first to the proxy's tool that, in turn, accesses the Web page requested by the client, inserts the data-logger into the Web page, and then returns the enhanced page to the client. The strongest aspect of this approach is the possibility of analyzing websites even if the evaluators are not administrators of the website to be studied because this strategy does not require changes in the source code of Web pages or access to Web server logs [103]. However, the proxy-based approach may result in a delayed response time in order to process the Web page and insert the data-logger code and rewrite links so they point to the tool's proxy. In addition, the code inserted into the requested page must deal with any error or incompatibility of the evaluated website. Finally, this type of approach also raises security concerns because this kind of logger can be used for malicious purposes to attack and gather private information such as passwords [31].

Capture at the client-side may be achieved through data-loggers inserted into Web pages or via specifically tailored Web browsers. An interesting characteristic is the highly detailed data available at the client-side. However, it is necessary to include the data-logger in all Web pages to be evaluated or to tailor the Web browser.

Regarding data analysis, critique, and adjustments, different approaches are proposed in the literature. Tools consider task models, heuristics, statistical metrics, Web Usage Mining, among others. The following section presents a description of relevant tools found in the literature considering the evaluation tool characteristics discussed previously.

5.2.2 Website automatic evaluation tools

WebVIP is a logger for formal tests. The vocabulary of events, which stands for the number of different event types, is restricted to a few events (i.e., press/hold keys, press/hold/move the mouse pointer, enter/leave a widget, and enter/exit the window). The environment configuration requires a local copy of the entire website being evaluated [77]. WET is another example of a logger for formal tests. It uses cookies to store logged data, leading to the reduction of the vocabulary of events due to storage issues [37]. These loggers represent the first efforts to capture client-side events.

WebRemUSINE is a tool that performs the automatic capture and analysis of website interaction logs in order to detect usability problems through remote evaluation. The analysis of logs is based on the comparison between the paths used by users and the optimum task model configuration. The data transmission is conducted by a Java applet component. The user must select the tasks s/he is performing so that the events captured can be related to the task selected by the user [83]. MultimodalWebRemUSINE is the latest version of the tool that aims to exploit the possibilities opened up by recent technologies to gather a richer set of information regarding user behavior. The tool allows traditional graphical logs to be analyzed together with the logs from webcams and portable eye trackers [85].

Google Analytics is an example of automatic capture and analysis tool. The default data source used by the tool represents page-views. The tool requires the evaluator to register him/herself and to insert a JavaScript code into the Web pages to be evaluated. It provides different report formats, allows actions to be registered as virtual page-views, and has a feature to register customized events at the client-side. These customized events in Google Analytics are events that can be named by the evaluator and triggered in any Web page component configured to communicate with the JavaScript data-logger (e.g., a Flash video or HTML event handler). However, the tool has a limit of logging 500 customized events per visit [43]. Google Analytics is the most popular evaluation tool, being used in 53.5% of websites [119].

WebQuilt is an automatic capture and analysis tool that uses page-view level logs as the data source. It uses a proxy-logger that mediates between users and Web servers and stores the communication between them [52]. MouseTrack is a proxy-based usability evaluation system that performs automatic client-side capture and analysis. It provides an online configuration and visualization tool that shows the mouse path followed by website visitors [7]. UsaProxy is a proxy-based usability evaluation system that performs automatic capture and analysis of client-side events. It uses JavaScript and focuses on usability tests [8]. WebinSitu is an enhanced version of UsaProxy that focuses on behavior comparisons between blind and sighted users [13]. WAUTER is a proxy-based Web usability evaluation tool. It employs a functional set of tools that automate the capture and analysis through the use of client-side logs and task models [9]. Web Usability Probe (WUP) is a proxy-based remote usability evaluation tool that considers formal use situations. The data source considered is client-side data on user interactions and JavaScript events. In addition, it allows the definition of customized events, giving evaluators the flexibility to add specific events to be detected and considered in the evaluation. The tool supports evaluation of any Web site by exploiting a proxy-based architecture and enables the evaluator to perform a comparison between actual user behavior and an optimal sequence of actions [17]. In some cases, proxy-based tools require reconfiguration of the user's browser or a proxy setup. Moreover, they may result in Web server processing overhead, due to additional requests/responses, or compatibility problems, which may occur when inserting JavaScript code into the evaluated Web pages.

Web Utilization Miner is a system for the discovery of navigation patterns in websites, which are represented in digraphs [113]. WebSIFT (Web Site Information Filter) is a system that aims at discovering knowledge from usage, content, and structure of websites [25]. LumberJack is a tool that processes Web server logs and uses the content and hyperlinks to build a model of user activity, applying clustering and a number of statistical analyses to the data [21]. DCW (*Descubridor de Conhecimento en la Web*) is a tool for discovering navigation patterns and association rules in server-side logs [35]. WebCANVAS is an evaluation tool that uses server-side logs. The tool applies clustering techniques to group similar navigation patterns and supports the visualization of theses clustered patterns [16]. These WUM tools have provided interesting contributions to the statistical analysis of usage and to the discussion on how to represent the outcomes of an evaluation in a summarized way. However, the logs used in these studies are captured at the server-side and thus do not include detail of interactions.

Although researchers have shown interest in this field and a number of automatic evaluation tools have been proposed, some gaps still remain. A tool can automate the usage capture and analysis, but this kind of tool requires a setup and specific client configuration. In addition, some tools can keep track of users' mouse movements but are unable to deal with usage data from the wide diversity of users [100]. Other techniques used to log or store usage data reduce the event vocabulary or the number of events captured in each session. Additionally, task modeling has been advocated by some authors [83, 85, 113], but the creation and maintenance of the task model can be time-consuming for developers. The tools should avoid requiring significant effort on the part of the participants in the evaluation (e.g., configuration of the browser, access to a bookmarked proxy link), as well as on the part of the evaluator. Table 5.1 summarizes the characteristics of relevant evaluation tools.

Evaluation tool	**Evaluators effort**	**Users effort**	**Data logged**
WebVIP	Copy of the entire website being evaluated and the insertion of code into each Web page element to be logged	Perform defined tasks	Restricted vocabulary of client-side events
WET	Insertion of the data-logger into the Web pages	Select the beginning and the end of tasks	Restricted vocabulary of client-side events
WebRemUSINE	Task model definition and its maintenance	Selection of the task being performed	Client-side events
MMWebRemUSINE	Task model definition and its maintenance	Indication of the task being performed via one modality	Client-side events, video, and eye tracks
Google Analytics	Insertion of code into each Web page element to log abstract events and virtual page-views	None	Client-side customized events, page-views, and virtual page-views
WebQuilt	Invitation for users to access the proxy every session	Access the proxy	Page-views
MouseTrack	Invitation for users to access the proxy every session	Access the proxy	Mouse events
UsaProxy, WebInSitu, and WAUTER	Invitation for users to access the proxy every session	Access the proxy	Client-side events
WUP	Invitation for users to access the proxy every session	Access the proxy	Client-side events and customized events
Web Usage Miner, WebSIFT, LumberJack, DCW, and WebCANVAS	None	None	Page-views

Table 5.1: Summary of evaluation tools' characteristics regarding their environment configuration, actions performed by participants, and level of detail of the logged data

5.2.3 A taxonomy proposal for website evaluation tools

Considering the scope of website evaluation tools, we propose a taxonomy that considers the interaction between participant and evaluator, as discussed in [85], and the automation type, as presented in [59]. The proposed taxonomy gathers previous contributions and extends them by adding the data source and effort level dimensions, as discussed in [103, 100]. The resulting taxonomy consists in the following dimensions:

1) Participant-evaluator interaction – refers to the interaction between evaluators and participants during an evaluation;
2) Data source – refers to the data source considered in the evaluation;
3) Effort level – refers to the effort required from the evaluator and from the participant to setup an evaluation scenario;
4) Automation type – refers to the automation characteristics of the tool.

The proposed taxonomy is structured as follows:

1) Participant-evaluator interaction
    a) Localization
        i) Remote
        ii) Non-remote
    b) Time
        i) Synchronous
        ii) Asynchronous
    c) Use
        i) Formal
        ii) Informal
2) Data source
    a) Web page data
        i) Structure (e.g., HTML page)
        ii) Content (e.g., text inside markup tags)
    b) User data
        i) Usage data (e.g., client-side log)

    ii) Questionnaire

3) Effort level

   a) Evaluator

      i) Model development/maintenance (e.g., task model/descriptions, user model)
      ii) Environment configuration (e.g., subscribe, logger insertion)
      iii) No action (e.g., use Web server logs)

   b) Participant

      i) Actions during the test (e.g., select task or indicate task start/finish, indicate that a critical incident occurred)
      ii) Action at the beginning of the test (e.g., access tool's proxy or acceptance)
      iii) No action (i.e., user is unaware of his/her participation in an evaluation)

4) Automation type

   a) Capture

      i) User expressions (e.g., via camera or eye trackers)
      ii) Ambience (e.g., geo location, luminosity)
      iii) Browser events (e.g., triggered by the user or by the system)
      iv) Customized events (i.e., single, sequenced, or composed)
      v) Page-views

   b) Analysis

      i) Visual reports
      ii) Statistical reports

   c) Critique

   d) Adjustment

      i) Content
      ii) Structure
      iii) Layout

Having presented the taxonomy, Table 5.2 classifies each of the tools described previously, providing a panorama of evaluation tools.

			Tools
1. Participant-evaluator interaction	Localization	Remote	14
		Non-remote	2
	Time	Synchronous	3
		Asynchronous	14
	Use	Formal	6
		Informal	10
2. Data source	Web page data	Structure	1
		Content	1
	User data	Usage data	15
		Questionnaire	1
3. Effort level	Evaluator	Model development/maintenance	4
		Environment configuration	8
		No action	5
	Participant	Actions during the test	4
		Action at the beginning of the test	6
		No action	6
4. Automation type	Capture	User expressions	1
		Ambience	1
		Browser events	10
		Customized events	4
		Page-views	7
	Analysis	Visual reports	11
		Statistical reports	4
	Critique		0
	Adjustment	Content	0
		Structure	0
		Layout	0

Table 5.2: The taxonomy instantiated for website evaluation tools surveyed.

In the Participant-evaluator interaction dimension, it is possible to observe that the remote and asynchronous evaluations are the most frequently considered by the tools. Regarding use, informal use is somewhat more frequent than formal use. This may be due to the fact that formal use is considered in many tools that capture detailed interactions while informal use is more present in WUM tools.

Regarding the Data source dimension, the literature review shows that usage data is most commonly considered. Moreover, structure, content, and questionnaires may complement usage data and could be combined in new evaluation tools. This combination is a promising direction revealed in this second dimension.

On the Effort level dimension, considering evaluator efforts to setup and maintain the tool, the attribute that is most commonly shared among the studied tools refers to the environment configuration required by evaluation tools that capture detailed data. When evaluators obtain data from Web server logs, the tool is classified as "no action". Effort level possibilities on the part of the participants are equally distributed among the three categories (i.e., actions during the test, action at the beginning of the test, and no action). Selecting tasks is related to evaluation tools that require task models or task descriptions, these being characteristics of tools that consider formal tests. Accessing proxy is related to tools following proxy-based architectures. No user action is frequent in WUM tools. Thus, a promising direction regarding the third dimension is to obtain detailed data, thus requiring some environment configuration. In addition, on the part of the participant, the less effort needed the better, except by the necessary condition of accepting to participate in the study, or accessing the proxy, making users aware of the evaluation.

Regarding the Automation type dimension, capture considering browser events is the most frequent, probably influenced by the JavaScript recent popularity and easy environment configuration in comparison to evaluations that require logging of user expressions or eye movements. The ambience data is promising especially when considering mobile applications. Concerning the analysis, graphical reports are commonly used, sometimes in conjunction with statistical reports. Moreover, since the survey produced by Ivory and Hearst [59], few tools provide critique features and, in the context considered in this work (i.e., website evaluation tools), none of the studied evaluation tools provide suggestions on how to treat the problems encountered. Likewise, none of the evaluation tools consider the adjustment feature, as recently proposed in [100]. Thus, critique and adjustment represent a potential for exploration in the fourth dimension. Other promising solutions are the capture involving customized events and the analysis via graphical reports, potentially in conjunction with statistical results.

Finally, the classification of surveyed tools presented in Tables 5.1 and 5.2 reveals a gap with respect to the combination of WUM techniques with detailed data. The next sections present the proposed tool that aims to follow those promising directions.

5.3 The Proposed Tool

WELFIT (Web Event Logger and Flow Identification Tool) is an evaluation tool that supports remote/non-remote, synchronous/asynchronous, and formal/informal tests. The data source considered is client-side log. Regarding the effort level to configure an evaluation, the evaluator is required to register and to insert the logger into Web pages; on the part of the participant, it requires the acceptance of the invitation to participate in the evaluation. Finally, the automation performed by the tool involves logging client-side events and generating graphical/statistical reports. The tool deals with the previously discussed shortcomings by avoiding the limit of client-side events captured, capturing all types of events triggered at the client device, by providing simple environment configuration, by summarizing usage patterns, and by pointing out usage incidents.

Paternò and Santoro [85] present some limits of client-side logging: special software must be installed on the client; there needs to be a mechanism for sending the logged data back to the team that wants to collect the logs; finally the software, in some cases, is platform-dependent, meaning that the software only works for a specific operating system or a specific browser. These drawbacks are all avoided in the client module used in the proposed approach.

5.3.1 Methodological basis

The proposed system for identifying interaction usage patterns is methodologically grounded in Organizational Semiotics (OS). OS is a discipline that deals with information and information systems, taking into account both technical and human aspects [117]. OS has a set of methods named MEASUR (Methods for Eliciting, Analyzing and Specifying Users' Requirements) for system modeling and requirements specification in software development [62]. The artifact of the MEASUR that guided the tool specification was the Semiotic Ladder (SL). It supports the analysis of information systems in six different information layers, contributing to the clarification of what is needed to produce a system encompassing its Information Technology (IT) Platform, but also considering social aspects regarding how the system is intended to be used. SL supported the system conception and helped in the elicitation of requirements and in clarifying the gaps existing in the literature review [103]. Understanding of both Human Information Functions and the IT Platform are especially relevant for considering individual needs in a context of diversity [104].

The basic requirement of the developed system is to capture and log user interface events, which is information commonly available in interactive systems, reducing the need for specific evaluation devices (e.g., eye tracking) that take for granted certain characteristics of the user population (e.g., sight). The SL containing all the elicited requirements for evaluation tools can be seen in [104].

5.3.2 Tool's functioning and environment configuration

The system has two main modules. The client module is responsible for capturing events at the client-side, iteratively compacting the data and transmitting the packages of logged data asynchronously to the server. The server module receives the data sent by the client module and stores them for future analysis.

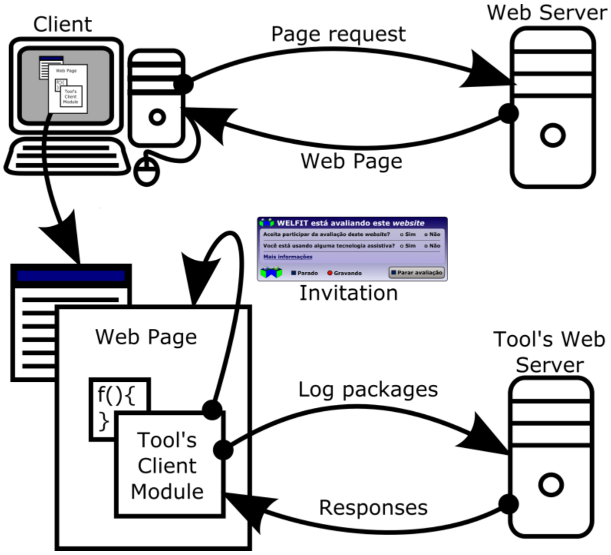
For the evaluator, the environment configuration requires the following steps:

1) The website administrator must register him/herself at the tool's Web administrative interface.

2) Once logged, the administrator must register the websites s/he wants to evaluate.

3) Once the website is registered, s/he includes the call to the JavaScript client-module in all website's pages that are to be evaluated.

Abascal and Nicolle [4] present important questions that HCI practitioners must be aware of when considering socially and ethically sensitive information systems. Thus, the tool requires the user to accept participation in the evaluation.



WELFIT está avaliando este *website*

Aceita participar da avaliação deste *website*? ○ Sim ○ Não

Você está usando alguma tecnologia assistiva? ○ Sim ○ Não

Mais informações

■ Parado ● Gravando

Figure 5.1: Overview of the actions performed in the invitation and data capture.

The data are captured according to the following steps (Figure 5.1):

1) User requests a Web page that uses the WELFIT.

2) The Web page containing the tool's client module call is retrieved to the user.

3) Client module requests session specific information to the tool's Web server, which also verifies whether the request is coming from a registered website. If the website is registered, then, as soon as the client module is loaded, the user receives an invitation to take part in the evaluation. The user is also asked whether s/he is using an AT, and if so, the session is flagged for further analysis.

4) If the user agrees to participate, the capture starts, and logs are sent via asynchronous communication to the server module.

5) If logs are properly stored at the server, the response notifies the client module that it can delete data just sent.

At the client module, as soon as a package containing the logged data reaches the configurable size limit, it is sent to the server. Thus, as soon as some data are recorded by the server, the evaluator can login and check the resulting usage graph, which is the digraph representing the UI usage in which each node represents an event triggered in a certain Web page element. The usage graph also can be seen as the combination of walks (non-empty alternating sequence of nodes and edges) representing what, where, and when users performed actions. In the usage graph a node is identified by its label, which is the concatenation of the event name and an identifier of the UI element where the event occurred. Moreover, each node counts on information regarding the total of sessions they occurred, mean distance from the root node, mean timestamp, among others. Algorithm 5.1 shows the procedure used to build the usage graph, how the approach used by the tool deals with session tracking through the use of session identifiers, and how the logged data is selected through data base reading operations.

The procedure presented in Algorithm 5.1 also indicates that the data source involves event streams occurred in a certain Web page, thus a usage graph can be built for a single session, for a cluster of sessions, or even for all the sessions occurred in a certain Web page. Detailed information regarding the usage graph definition and proposal can be found in [105].

The client-server communication is performed according to the following policy. The client module (Figure 5.2) asks the server to store packages of logs, and the server answers the requests. At a defined clean up cycle, the server responses are checked by the client module. The clean up cycle used is 10 seconds or 10 accumulated packages (i.e., not deleted at client due to the lack of server response). When the client receives the server's answer, the module stores the received answer. In the next clean up cycle, the confirmed packages are deleted, the error packages are resent, and, if the denied message is received (which occurs if the website is not registered at the server), the client module halts.

```
input  : Collection of event streams occurred at a certain Web page. In each event
         stream, events are ordered by timestamp.
output: Usage graph structure. Vertices represent events.
G = new graph;
G.addVertex( Start );
G.addVertex( End );
foreach stream S in event streams collection do
    PreviousVertex = Start;
    Distance = 0;
    S = Start + S + End;
    foreach event E in event stream S do
        V = new vertex;
        V.setName( E.targetTag + ‘-’ + E.targetId ‘-’ + E.type );
        if ( !G.containsVertex( V ) ) then
            V.setN( 1 ) ; V.setMeanDistance( Distance ) ;
            V.setMeanTimestamp( E.timestamp ) ;
            G.addVertex( V );
        else
            V = G.getVertex( V.getName );
            V.getN().increment();
            V.setMeanDistance( ( ( V.getN -1 )* V.getMeanDistance + Distance ) /
            V.getN );
            V.setMeanTimestamp( ( ( V.getN -1 )* V.getMeanTimestamp +
            E.timestamp ) / V.getN );
            G.setVertex( V.getName, V );
            if G.containsEdge( PreviousVertex, V ) then
                G.getEdge( PreviousVertex, V ).getWeight().increment() ;
            else
                G.addEdge( PreviousVertex, V );
            end
            PreviousVertex = V;
            Distance++;
        end
    end
end
```

Algorithm 5.1. Usage graph construction

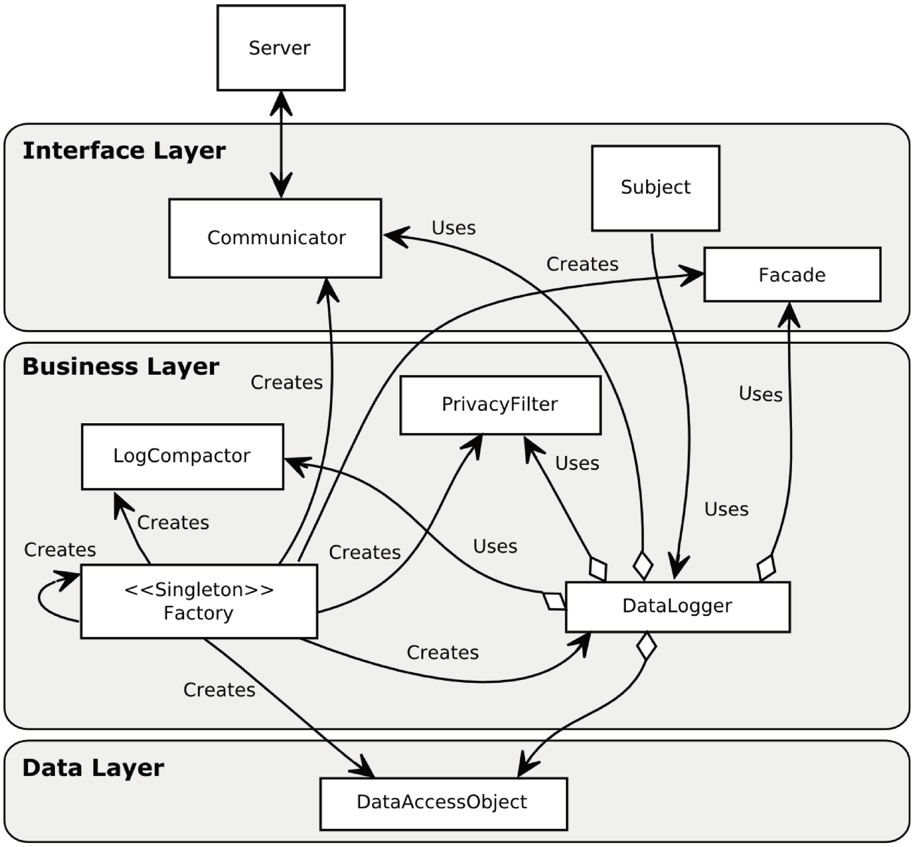


Figure 5.2: Client module overview representation of its architecture and classes.

Another halt situation occurs if the user's bandwidth connection does not support the amount of recorded data, which will cause the limit of 20 accumulated packages to be reached. The specification details and rationale of the client module can be found in [100]. The package size, clean up cycle, and limit of accumulated packages are all configurable and may differ when considering other devices or modalities other than the ones tested in this work. The numbers presented are based on results of the use of the proposed evaluation tool in the Web.

5.3.3 Indentifying Web usage patterns

The presented tool considers SAM (Sequence Alignment Method) for measuring distance between event streams. A SAM-based heuristic is applied to point out usage incidents. The SAM-based heuristic was proposed and formally defined in [105]; in this work it is presented as a tool's feature.

SAM allows grouping sessions considering the structural information present in event streams. SAM is a distance-based measure that gives a score representing the amount of operations necessary to transform one sequence (x) into another (y) [49]. The SAM distance is calculated as the sum of the number of weighted operations.

The operations used to transform $x[1..m]$ into $y[1..n]$, using an auxiliary array z, are as follows [26]:

(1) Copy a character from x to z by setting $z[j] = x[i]$ and increment both i and j.

(2) Replace a character from x by another character c by setting $z[j] = c$ and then increment both i and j.

(3) Delete a character from x by incrementing i, but leaving j alone.

(4) Insert the character c into z by setting $z[j] = c$ and increment j, but leave i alone.

(5) Twiddle (i.e., exchange) the next two characters by copying them from x to z but in the opposite order; we do so by setting $z[j] = x[i+1]$ and $z[j+1] = x[i]$. Then, we set $i = i + 2$ and $j = j + 2$.

From these operations, the SAM distance used is calculated as the sum of the number of operations and each sum is multiplied by a respective weight (5.1). The weight of replace operation was assumed to be less than or equal to the sum of deletion and insertion weights because their results are the same (5.2). For the same reason, the twiddle operations weight is less than or equal 2 times the weight of the replace operation (5.3). For example, the $d_{SAM}('abcc',' abd') = 3$ because it requires a replacement and a deletion. The SAM distance is considered in this study because it maintains sequence information, which is the case when analyzing event streams, since the order in which events occur is important information when evaluating UIs. In addition, event streams' length differs and this information is also an important variable to consider. For example, an event stream containing a significant number of keyboard events until reaching a certain link can be compared to another event stream containing fewer events using a pointing device, which may indicate that the UI being evaluated does not offer an efficient way of navigating through keyboard. Finally, the weights used in our application of SAM are the following: $w_d = 1, w_i = 1, w_r = 2$, and $w_t = 3$, so that the replace operation is equal to a deletion plus an insertion, and twiddles are $\frac{3}{4}$ of 2 replacements.

$$d_{SAM}(S_1, S_2) = w_d D + w_i I + w_r R + w_t T \tag{5.1}$$

$$w_r \leq w_d + w_i \tag{5.2}$$

$$w_t \leq 2w_r \tag{5.3}$$

To classify whether sessions are coming from AT users or not, the edit distance for all pairs of sessions is calculated for each page. As the system calculates a new distance between two sessions, it stores the result in the tool's database. Once all distances are calculated, a divisive approach is used to bisect the sessions. Then, the cluster with more users that affirmed to be using an AT is flagged as the AT behavior group.

Considering the properties of SAM, a heuristic was defined to highlight significant distance differences to a certain digraph node. This allows an approximate comparison of distances between more than two sequences at once and at every node with an out degree greater than one. The heuristic is based on a metric that aims to evaluate the distance between nodes (i.e., the number of events triggered between every event). The intent of the heuristic is to point out transitions of the usage graph that may represent a usage incident, using the fact that subsequences of different lengths require more operations (e.g., insertion and deletion) when using SAM. In the same sense of the heuristic, Winckler et al. [128] comment that, in well-designed interfaces, the user can quickly reach his/her goals by performing few actions and that if there are cyclic patterns, the evaluator must infer that something is going wrong. Either the users did not accomplish the proposed task or a usage incident occurred.

In the proposed tool, WUM is used to group walks (i.e., non-empty alternating sequence of nodes and edges) in order to identify usage patterns and UI design problems. Server log studies use page-view as the basic navigational unit or digraph node. In the proposed tool, the navigational unit used is the concatenation of an event, a page element, and an ID or name. For example, a representation of a click event at an anchor tag (i.e., link) identified as logout is "click-a-logout". Additional details about the data summarization and usage graph can be found in [105].

Bearing in mind the properties of the mentioned heuristic, consider a digraph G representing events occurred in UI elements and a certain node of G, say u, which has an average distance d_u from the root node, i.e., the beginning of the usage. Additionally, let u's neighbors be $v_0, v_1, ..., v_k$, and the respective average distances be $d_{v0}, d_{v1}, ..., d_{vk}$. Thus, if $d_{vi} < d_u$, then the in degree of v_i is greater than 1, meaning that some alternative walk to v_i exists that does not have u as precedent node and has a smaller average distance. Thus, the walk from the root to u is not the shortest, indicating a performance difference and that u is part of a walk that may be part of a usage incident (Figure 5.3).

To represent these performance differences, a minimum and a maximum value are applied. They follow the 3-sigma rule and are used to define u's neighbors according to the following:

- If $d_{vi} > d_u + 2s$ (where s involves u's neighbors), then the node is marked as a potential usage incident (i.e., warning node).

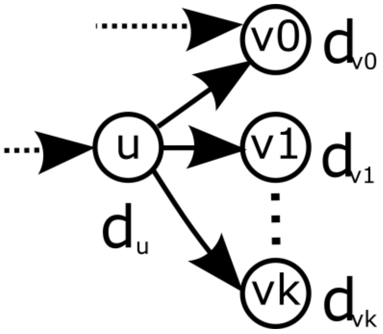


Figure 5.3: Properties that inspired the SAM-based heuristic.

- If $d_{vi} < d_u - 2s$, then it represents a shortcut to other walks with greater average distances (i.e., shortcut).
- In other cases, we consider that the node does not have significant information regarding performance differences.

Considering the presented approach, the following example illustrates how event streams are presented by a usage graph and how the SAM-based heuristic is applied by the tool. First consider these event streams (differences between S_1 and S_2 are in bold):

- $S_1 =$ "mousemove-body, mousemove-body, mouseover-a-logout, mousemove-a-logout, mousemove-a-logout, **mouseout-a-logout, mousemove-body, mousemove-body, click-body, click-body, click-body**";
- $S_2 =$ "mousemove-body, mousemove-body, mouseover-a-logout, mousemove-a-logout, mousemove-a-logout, **click-a-logout**".

S_1 represents one of the patterns found that indicates that a user performed clicks on a non-clickable element, in this case, the document body. S_2 represents one of the expected event sequence for clicking at the logout link. The resulting $d_{SAM}(S_1, S_2) = 7$, since it requires one replacement and five insertions operations. Finally, the resulting usage graph for S_1 and S_2 is presented in Figure 5.4.

The tool was developed to require low effort in the environment configuration, to use client-side capture during real use of the widest possible audience, to guarantee user control, to return summarized information of captured sessions, and to not require specific devices or events vocabulary. In addition, it does not depend on a user's action to perform primary functions (e.g., send data to the server). It integrates A&U in website evaluation and uses WUM techniques to identify usage patterns without depending on task models. These characteristics were guided by requirements presented in [103] and revised in [104].

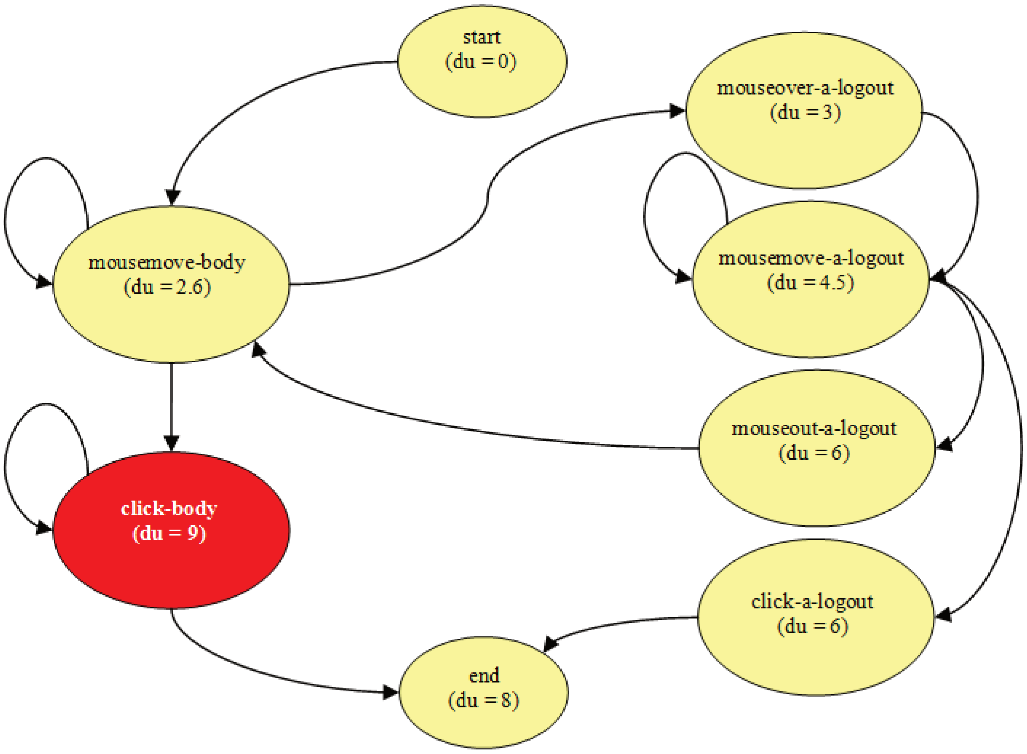


Figure 5.4: Usage graph resulting from S_1 and S_2. Nodes are labeled considering event and element name; d_u represents the mean distance from the root (start) node. The highlighted node is the usage incident indicated by the SAM-based heuristic.

5.4 Tool Validation

The validation of the WELFIT considered a study involving real usage and how the tool is used remotely during informal use; for a non-remote study of a formal situation where the tool was applied, see [104].

The goal of this case was to identify usage patterns and the accuracy of the SAM-based heuristic. Beyond that, this case involved the test of the hypothesis that event streams triggered by AT users and non-AT users differ in the event types that compose them.

5.4.1 Method

Participants: The study counted on 180 participants. They were visiting the studied website and accepted the invitation of being part of the evaluation. The website counts on information of a research group website . The website was chosen because part of its audience uses AT, allowing us to address case's goal and hypothesis.

Materials: The data set used in this case was captured by WELFIT during a 15 month period of logging the interaction of users with the studied website. The data was captured remotely and during informal use (i.e., participants were not asked to accomplish a predefined task).

Design: The variables considered are the acceptance, the use of AT, and the types of events contained in the event streams. For evaluating the SAM-based heuristic, the usage graph and its nodes were the variables considered.

Procedure: All visitors that first accessed the evaluated website after the inclusion of the client module were invited to be part in the evaluation. In the invitation they also answered if they were using any AT. To evaluate the heuristic 10% of the 531 URLs related to the logged sessions were randomly chosen. In each usage graph, nodes present in less than 50% of the sessions were cut from the usage graph (i.e., rules with support greater than 50% were evaluated), and each warning node present in the reports was mapped to the UI using the ID as a reference in order to verify false warnings. This rationale aimed to verify patterns and does not mean that the cut nodes are not meaningful. This was only a strategy for analyzing frequent usage patterns.

5.4.2 Results

A total of 205 users were invited. A sum of 180 accepted to participate in the evaluation, of which 30 informed that they were using AT (16.67%). Twenty five of the original 205 users declined to participate (12.19%), and one of them indicated being an AT user. Table 5.3 summarizes acceptance and participants' responses. Bearing in mind that the evaluation was conducted remotely, during real use, and that the only question to answer at the beginning of the evaluation was if the participant was using AT or not, it was not possible to gather gender or demographic data.

Acceptance / Uses AT	Use AT	Do not use AT	Total
Accepted	30	150	180
Did not accept	1	24	25
Total	31	174	205

Table 5.3: Participants' acceptance and use of AT.

During the 15 month period, 246 sessions were logged: 30 were related to AT users and 216 to non-AT users, 16.67% of them were returning visitors. The sessions included a total of 440,056 client-side events and 531 different URLs across the evaluated website.

In order to analyze the resulting data related to the event streams composition, as suggested in [126], the Shapiro-Wilk normality test was applied in the data related to the 246 logged sessions. The normality test indicated that the data set does not follow a normal distribution. Hence, the Wilcoxom Rank-sum test was applied in order to verify significance difference in the type of events that compose the logged event streams. Results indicate difference of distribution involving the length of event streams coming from AT (M = 68.83, SD = 127.20) and non-AT users (M = 116.39, SD = 235.38), W = 4530.5, p = 0.01, difference in location = 12. Regarding the presence of events more related to a certain device, results indicated the following results for the presence of events originated by the mouse of AT (M = 48.35, SD = 112.79) and non-AT users (M = 86.49, SD = 191.29), W = 4596, p = 0.01, difference in location = 7.99. No significant differences were found regarding events triggered by keyboard or by the browser. Remarking the initial hypothesis that event streams triggered by AT users and non-AT users differ in their composition, it is possible to verify that length of the event streams and the presence of mouse event are promising directions on classifying event streams without requiring the participants' input.

Considering the SAM-based heuristic, results indicated that the heuristic achieves its goal of discovering performance differences and supporting evaluators in understanding how an evaluated website is really used. The analysis of true positives contained in the sample revealed that the heuristic points actual usage incidents in more than 65% of the cases (M = 73.84, SD = 35.07), V = 947, p = 0.01.

Usage patterns and usage incidents found in the usage graph sample are:

– AT users in the participants group access the accessibility toolbar containing skip links, font-size controls, etc., as one of the first navigational elements. One possible explanation for this behavior is that low vision users need to adjust the Web page's font-size or contrast before using it;

– At the homepage, WELFIT highlighted repetitions of mouse movements and clicks over the logo of the website, representing the exploration or the return to the main page. The usage incident found may be related to the fact that the logo is linked to the homepage itself. It is possible that the change of the cursor's format to the hand is influencing the usage by people with low vision. This pattern was more present in sessions of AT users, who were influenced by the lack of contextual information when a magnifier tool is used in the logo area;

– Another usage incident found was that users performed repeated mouse events over the search text input box. This can be related to the lack of contrast between the background of the input text and the background of the Web page;

– Breadcrumb is a navigational element commonly used in the session samples studied; however, clicks were also identified on regions of the breadcrumb that are not links. This is possibly related to the fact that breadcrumb links and text have almost no contrast (they differ only in color, from black to dark blue);

– One usage incident found was related to the interaction with the tool's widget itself. One participant had problems when trying to stop the logging. This incident guided the redesign of the widget to allow users to easily stop the event recording at anytime.

5.4.3 Discussion

The results regarding event streams composition can be useful for evaluators and miners. It was possible to verify that the event streams composition vary in a noteworthy way; standard deviations indicate this. For evaluators, they help on identifying and removing accessibility barriers for users that navigate mainly through keyboards. For miners, they help on the task of characterizing event streams through WUM algorithms for classifying or grouping event streams. In this regard, we performed a pilot experiment using the Weka for classifying the event streams considering length of event streams and proportions of mouse, keyboard, and system events. The result obtained the correct classification of 83.14% of the event streams (i.e., classifying them as AT or non-AT users). This pilot study used classification via clusterization, applying the Sequential Bottleneck algorithm; 66% of the event streams were used for training the algorithm and 34% for testing.

From the graphical reports representing usage patterns, it is possible to rebuild the interaction of users with the UI. Due to the detailed data captured, WELFIT offers evaluators a feature that allows them to cut nodes from the usage graph. For instance, to emphasize repeated actions and relevant data, the evaluator can choose to see only events triggered by the majority of participants, keeping only events present in more than 50% of the sessions in a certain URL. Using the WUM terminology, this means that evaluators can select the support for the association rules found in the event streams. In the usage graphs, the proposed heuristic revealed noteworthy performance differences, indicating events and UI elements that deserve consideration.

False positives were frequently related to repeated mouse movements over text elements. One explanation of the presence of these movements is that some users use the mouse to point to sentences being read.

Finally, if the evaluator chooses to cut the less representative nodes from usage graphs, many connected graph components may appear. This may help evaluators to identify patterns, but this may also render it harder to inspect the warnings because some context information is lost when reducing the graph. This reinforces the fact that all the information present in event streams is of value for evaluators in providing the overall context in which a pattern occurred. One way of evaluating these warnings in a more contextualized way is to analyze clusters of sessions with high cohesion.

5.5 Conclusion

The Web is far from being genuinely accessible. Evaluating Web pages' source code is not sufficient for developing accessible or usable websites. Moreover, the use of server-side logs as data source does not allow evaluators to obtain detailed information about how a website is used. On the other hand, client-side logs provide detailed information regarding which actions were performed and which UI elements were used. The evaluation tools reported in the literature have some limitations, especially with regard to the combination of capturing detailed interactions, applying WUM techniques on evaluation tools, and considering data that are present in sessions of users with or without any disability.

The tool presented in this work opens up new possibilities for the application of WUM techniques with client-side event logs. With the continuous evaluation supported by WELFIT, it is possible to identify Web usage patterns mediated by AT during real use. It combines data capture, data analysis, summarization, and presentation of detailed usage data to create an approach that is, to the best of our knowledge, not present in the literature or as a tool available to the HCI community. Usage patterns may show sequences of actions that provide data on how the UI is used. These patterns represent a potential source for improvements in the UI design through providing personalization features such as the automatic resizing of UI elements, the reordering of elements, and changes in coloring, to name a few.

With the case presented, it was possible to verify that the tool achieved its goals on: remote, informal, and asynchronous participant-evaluator interaction; gathering usage data; the effort level required on the part of the evaluator and the participant; and on the automation types considered, i.e., capturing browser events and supporting analysis via graphical and statistical reports. The captured data helped in the evaluation of the WELFIT approach. The proposed heuristics have shown a promising direction for dealing with usage incidents present in event streams captured during real use. In addition, the characterization of event streams could help in applying WUM algorithms.

The main contributions of this work are the conceptual proposal, specification and implementation of a tool to identify the usage patterns of websites considering detailed observational data related to real use. In the context of Web Accessibility, the proposed tool represents a promising direction of investigation, considering the characterization and mining of event streams related to AT users, which may facilitate the identification of accessibility barriers and support the examination on how AT users perform real tasks.

5.5.1 Limitations of the presented work

The study counted on people who volunteered in participating; they were not randomly selected. However, those participants were representative users of the evaluated website.

Another possible limitation is the consideration of only 10% of the logged URLs in the analysis. The random selection of the analyzed usage graphs tried to compensate this issue.

A general open question is what information to consider in the invitation. Results support the rationale of keeping it as short as possible. However, the inclusion of other questions as "what is the AT being used?" would enrich results and support analysis of other aspects.

5.5.2 Future work

WELFIT focuses on the evaluation of websites. Nevertheless, the main solutions implemented can be applied to other domains that consider a different target audience or use different devices. Recording client-side events and sending logs to a server are the basic mechanisms required to implement this type of tool in other domains. For example, it can be applied to evaluate how users use smart phones, video games, or digital television. It can also be applied to multi-modal information systems, which would require the development of a modality-dependent client module.

We also expect to use the characterization pointed out by this work to infer whether the participant is using AT without having to ask directly, allowing us to improve the invitation, by inserting a question of what kind of AT is being used. Moreover, we expect to involve more users and websites of different domains in follow up studies in order to identify the impact of the evaluation in more heterogeneous contexts.

Future work involves the use of usage incidents pointed out by the tool to automatically adjust the evaluated UI. Also the detection of false positives will be improved in order to fine tune the heuristic's accuracy.

5.6 Acknowledgments

We would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (PROESP/ CAPES) and *Fundação de Amparo à Pesquisa do Estado de São Paulo* (grant #2009/10186-9) for supporting this research. We would also like to thank the Institute of Computing of University of Campinas (IC/UNICAMP) and *Núcleo de Informática Aplicada à Educação* (NIED/UNICAMP) for the infrastructural support, as well as our colleagues from the Todos Nós, e-Cidadania, and InterHAD research groups. Finally, we would like to thank Diana Milena Galvis Soto for the suggestions on how to improve the data analysis.

Chapter 6

Self Tailorable Websites: a Contribution to the Design for All

Vagner Figueredo de Santana (vsantana@ic.unicamp.br) and
Maria Cecília Calani Baranauskas (cecilia@ic.unicamp.br)
Institute of Computing, University of Campinas (UNICAMP),
Albert Einstein Av., 1251, CEP: 13083-852, Campinas, SP, Brazil

Abstract: Due to the fast growing of the Web in the last decades and to the interactivity made possible by Web 2.0 technologies, government services, entertainment, and education are more and more available in the Internet. However, this growing has not been followed by considerations of the different capabilities of all users. Although there are techniques and approaches towards the Design for All, issues related to different user needs may emerge in different and not foreseen usage contexts. Self tailorable UIs represent a promising step towards promoting accessibility and usability in the Web by considering different usage contexts. Hence, this work reviews different approaches and proposes a solution that considers detailed observational usage data in order to identify users' behavior and to adjust the UI accordingly. The proposed approach supports continuous evaluation of websites to eliminate accessibility barriers and usability problems. From the results obtained by combining client-side event logs and Web Usage Mining techniques, the self tailoring approach may represent a step towards the Design for All.

Keywords: Tailoring, self tailorable websites, user interface evaluation, usability evaluation, accessibility evaluation, Web Usage Mining.

6.1 Introduction

The Web supports services from a wide range of areas and is becoming an essential part of people's life in our society. Examples that show this growing are the wide use of the Web on communication, entertainment, and commerce. According to Abascal and Nicolle [4], computers contribute on increasing the social inclusion and autonomy of people with disabilities, giving them access to education, work, information, leisure, etc. Hence, supporting the autonomy of people with disabilities is fundamental for a more democratic society.

Evaluation is a way of locating accessibility barriers in websites, to adequate them to accessibility and usability (A&U) requirements. Due to the large data volume resulted from evaluations, automatic tools have been proposed, enabling different types of data analysis and allowing the integration of evaluation activities along the development life cycles.

Continuous evaluation is aligned with the Web 2.0 development paradigm. Often, systems following the Web 2.0 paradigm label their production versions as 'perpetual beta' version, due to their constant evolving characteristic. O'Reilly [81] defines 'perpetual beta' as the version of a Web system that is used while it is under development; this version, constantly available for users, counts on new features added monthly, weekly, or even daily. Examples of applications that follow this concept are Gmail® and Flickr®.

Adding value to websites does not involve only offering huge amounts of data, but allowing easy access to the required information, at the right moment, and in the most adequate way [88]. Thus, interest in the analysis of users' behavior in the Web context is timely.

The study of users' behavior involves a number of variables. When the goal to be achieved is accessibility, the complexity of the usage context involving Web browser and all its variables is increased by the inclusion of another component, the assistive technology (AT). The Web Accessibility in Mind [56] conducted a study involving 1121 screen reader users in order to highlight how users configure and use AT. The referred survey indicated that AT users count on different configurations of hardware and software (e.g., desktop, laptop, cell phone, BrailleNote®, PacMate®, and personal digital assistant), different screen readers are used (e.g., JAWS®, Window-Eyes®, NVDA®, and VoiceOver®), and 90% of users perform some specific configuration in the screen reader. Considering outcomes of the WebAIM's study, solutions towards the Design for All should consider different configurations and preferences.

In a scenario involving different abilities and limitations even websites developed considering A&U may be part of challenging contexts of use. According to Abascal and Nicolle [4], due to the diversity of users' characteristics, it is almost impossible to anticipate all users' needs during project phases. All these possible contexts of use are hardly covered in tests performed in controlled environments. Thus, the continuous evaluation considering observational data is a way of considering these contexts of use as they occur. In the same direction, Mikroyannidis and Theodoulidis [69] say that websites should be adaptive by adjusting their structure and design considering the way they are used. Moreover, a self adaptive website should improve the usability for all users.

WELFIT (Web Event Logger and Flow Identification Tool) is an evaluation tool [100] that supports A&U in this regard. The tool indicates usage incidents that have possibly occurred during a real usage; a usage incident reflects an accessibility barrier, a usability problem, a missing functionality, or other ways in which a system fails to meet user needs. Client-side event logs are analyzed according to statistics metrics, without depending on task models or specific grammars. This work extends the previous version of WELFIT by adding a self tailoring feature based on Web Usage Mining (WUM) techniques. This new version of the tool will be referred from now on as WELFIT 2.0.

The main goal of WELFIT 2.0 is to reduce usage incidents by automatically identifying them from logged data and tailoring the UI accordingly, by continuously experimenting adjustments. Throughout the text, when the term experiment refers to adjustments, it means that a piece of JavaScript code is available (is being experimented) for participants and can be requested by the client module of the WELFIT 2.0. From the continuous evaluation and improvement of the UI one expects to promote the Design for All. In addition, as Pierrakos et al. [88] have shown, users feel more comfortable when using websites that recognize them as individual instead of generic visitors.

To achieve effective tailoring a rich data source is needed [38], corroborating the rationale of using client-side event data when tailoring websites to user needs. It is worth noting that the objective considered in this work is to improve a design solution without mischaracterizing it. For example, if data captured during a website visit reveal that before a successful click there were many pointing movement over the link, fact that allows evaluators to infer that the link is small or hard to point at, thus the UI may not be adequate for one or more users. Consequently, a possible adjustment would be to increase the size of the referred link so that future observations could verify whether the adjustment is valid for new logged data.

From a practical point of view this work involves the development and validation of WELFIT 2.0, which offers the self tailoring feature for websites. This new version has the following functions: identify usage incidents related to A&U, modify the evaluated website, and verify whether the adjustments applied were well succeeded. The approach involves applying pieces of JavaScript code in order to adjust the website UI. Adjustments are applied for clusters of one or more users. Moreover, attributes involved in adjustments are: ordering, size, and color of UI elements. The focus on using JavaScript command is to reuse them, since according to López-Jaquero et al. [63], most of the adaptive systems apply hardcoded adaptations, which makes hard of reusing the adjusted code. WELFIT 2.0 uses norms to provide a feature that allows the code to refer to non concrete elements, enabling the reuse of adjustments represented in the norms.

The research question considered in the work is: Does continuous evaluation based on users' behavior turn UI more tailored to users?

Finally, one expects to show that, from the self tailoring approach, it is possible to improve the UI design and continuously evaluate the UI designs that aim at A&U. The work is organized as follows: section 6.2 presents the literature review regarding evaluation of UI, concepts, and disciplines involved; section 6.3 shows the theoretical and methodological references that grounded this work; section 6.4 details all the steps performed in the self tailoring approach proposed; section 6.5 specifies the experiment design that supported the analysis of the adjustments; section 6.6 highlights the results obtained so far; and section 6.7 discusses the outcomes and points to future directions.

6.2 Background

There has been an increasing interest from academia and from industry in the use of UI event logs in evaluation tools. These logs allow the identification of the way users behave while navigating through websites and help in the understanding of the UI elements used. Server logs and event logs result in a noteworthy volume of data. A promising discipline counting on techniques to deal with these data sets is Data Mining. Hence, this work combines UI evaluation and Data Mining ideas to propose a self tailoring approach that aims at improving A&U. The next subsections present the literature review in which this proposal is grounded.

6.2.1 Concepts and disciplines involved

Web Accessibility means that people with different types of limitation can perceive, understand, navigate, interact, and contribute with the Web. Accessibility barrier is anything that makes difficult or impossible for people with disability to use the Web [125]. Usability, in sum, is the capacity of a product to be used by specific users to achieve certain goals with efficiency and satisfaction, in a certain context of use. Moreover, context of use involve users, tasks, equipments (hardware, software, and other materials), physical and social environment in which the product is used [58]. Usability problems can be defined as UI aspects that reduce the usability of the system for users [92]. Critical incident is a concept presented by Flanagan [39] and applied by Hartson et al. [48] in the area of Human-Computer Interaction (HCI) in one of the first works comparing local and remote usability evaluations. In that work the authors used the expression critical incident to represent a problem of usability, missing feature, and other ways in which a system fails if it is not adequate to users' needs. Revisiting the term critical incident used in remote usability evaluations as in [19, 48], in this work the term usage incident is considered a critical incident or an accessibility barrier. The rationale to use the term usage incident is also to avoid conflicts between terms considered in the critical systems field.

The access to Communication and Information Technologies (CITs) depends on our understanding about the ways these technologies are used by all. The combination of A&U is meant to reflect the idea that one may contribute to the other and vice versa. For example, access keys can be used as shortcuts to certain UI elements improving accessibility for blind users that use screen readers, but these access keys can also help users without vision impairment on using the same UI elements in a more efficient way.

According to Abascal and Nicolle [4], if services are not accessible, they are useless for people with disabilities; if services are accessible, it is also of key importance that users can accomplish tasks, easily and efficiently. The authors also comment that bad designs are not prejudicial only to people with disability, but to everyone. Thus, supporting accessibility in the Web does not exclude any user and extends the concept of usability [101].

When evaluating a UI with users, the setting can be informal, when the evaluation requires users to freely use the UI, or formal, when the evaluation requires participants to execute specific and predefined tasks. The automation of evaluation tools may involve: capture (i.e., logging the usage data), analysis (i.e., identification of problems), and critic (i.e., suggestions on how to improve the evaluated UI) [59]. In addition, it may involve adjustment (i.e., the elimination of identified problems) [100].

Evaluation of UI can be local, when the participant and evaluator are in the same place (e.g., a UI evaluation lab), or remote, when participant and evaluator are in different places. Moreover, evaluation can be considered synchronous (evaluator and participant need to work on the evaluation in the same time) or asynchronous (there is no need for evaluator and participant to work on the evaluation at the same time).

The combination of asynchronous remote evaluation with informal use is interesting to the context of website evaluation because avoids biases in the use of UI and consequently in the data gathered. It is also a way of enabling the number of sessions to scale. Another point in favor of this combination is that, according to Rubin [96], tests in controlled environments are artificial and may influence results. In addition, the variety of needs and the wide diversity of physical, sensorial, and cognitive characteristics of users make the UI design very complex [4]. Lastly, this combination is even more important when the evaluation considers websites, since it is hard to replicate configurations of hardware and software, especially when users are using UIs with AT support [13].

Currently there are three main solutions for gathering usage data of websites: server-side capture, proxy-based capture, or client-side capture.

- Server-side capture is performed by using Web server logs. It is the main data source considered in studies involving WUM [88]. Logs captured at the server-side have limitations. First, the effectiveness is strongly affected by the data validity, since server-side logs do not capture access to Web pages stored in cache; for example, if a user clicks on the back button and the retrieved content is in the cache, then this action will not reach the servers and thus will never be logged. Second, interpreting actions of an individual is hard because methods considered to capture data at server-side are not designed for gathering useful data regarding usability [38, 85]. The identification of users' sessions is more complex. Logs captured at server-side does not count on detailed information concerning users interactions with the UI being evaluated, it counts only on visited Web pages [103]. The strong point is the low cost, since it is a natural product of Web servers functioning.

- Proxy-based capture is an approach in which the capture module mediates the communication between the user and the server so that all users' requests are first sent to the proxy that, in turn, accesses the requested page, inserts the capture module in it and then returns the enhanced Web page to the users that requested it. The strong point is the possibility of analyzing websites even when evaluators are not administrators of the websites being evaluated, since there is no need of hard coded change in the Web pages [103]. However, there is the possibility that the overhead resulting from the manipulation of requests and responses results in a response delay for users. There is also the possibility of attackers to use the proxy-based capture to perform man-in-the-middle attacks. Another important requirement in this approach is that, if the capture module uses an interpreted programming language that is also present in the evaluated Web page (e.g., JavaScript), the code used in the evaluated website can interfere with the capture module functioning. This means that the code of the evaluated website and the capture module must be compatible and free of errors.

- Client-side capture can be performed via capture module inserted in Web pages or via specific designed Web browsers. The strong point regarding client-side capture is the level of detail that is possible to obtain; on the other hand, when a capture module is used, it requires the insertion of the module into all the evaluated Web pages. Moreover, in remote evaluations there is a need of sending the logged data to the evaluators, often the tool's server. According to Fenstermacher and Ginsburg [38], the client-side capture makes easier the association of actions to the users that performed them. Pierrakos et al. [88] argue that the data captured at client-side are more liable than the data captured at server-side, since the logs captured at the client-side are not affected by cache access.

Interface events are a natural result of the usage of UI based on windows and their components (e.g., mouse clicks, key strokes). From the possibility of recording these events and the fact that they indicate users' behavior during UI usage, they represent an important data source for tasks involving UI evaluation, and include data representing performed paths, repeatedly triggered events, time spent to perform certain actions, etc [51].

Data Mining is useful for the analysis of huge observational data sets in order to find non trivial relationships and to summarize them in such a way that they can be easily understandable [120, 46]. Web Mining was proposed as an area for unifying Data Mining methods applied on Web data. Hence, Web Usage Mining (WUM) is the process of discovering and interpreting how users access the Web via mining usage data. Lastly, WUM is recognized as a valuable source regarding ideas and solutions related to Web personalization [88, 111]. Figure 6.1 shows an overview of WUM process and Table 6.1 presents an overview of terms related to UI tailoring.

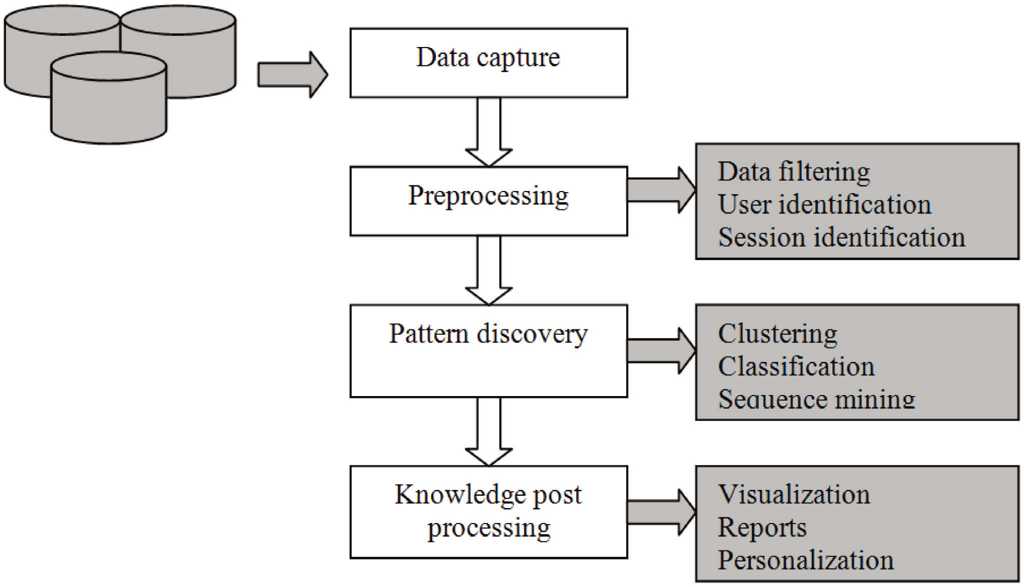


Figure 6.1: Overview of WUM process; adapted from [88].

Terms	**Definition**	**Can be divided in:**
Personalization [88]	Improves the experience of a visitor by presenting the information s/he wants, in an appropriate way, and at the appropriate time. It includes methods and techniques to provide users with a website with added value.	1. **Memorization:** Registry and recuperation of user information; 2. **Orientation:** Support the user to quickly obtain the information s/he wants; 3. **Customization:** Modifications applied to Web pages in terms of content, structure, and layout, considering users' knowledge and interest; 4. **Task support:** Execution of certain tasks on behalf of the user.
Personalization [70]	Any action that adjusts Web experience to a particular user or to a group of users.	
Individualization [76]	The user experience is adapted to each user need.	1. **Customization:** The user informs to the computer system what s/he prefers to see; 2. **Personalization:** The computer changes its behavior to adequate itself to the users' interests.
Tailoring [71]	Adaptation of information systems to specific practices of developers, end users, or group of users.	1. **Customization:** Modify the presentation of objects or edit attributes by the selection of predefined values; 2. **Integration:** Creation or recording of a sequence of actions that result in a new functionality, stored within the application as a component or a command; 3. **Extension:** Improve the functionality of an application by the insertion of new code.
Self-adaptive [69]	Improve the structure and design by learning how the UI is used.	
Adaptive UI [66]	Capable of considering the usage context and (automatically) react to context changes in a continuous way, changing presentation, content, navigation, or even behavior.	

Table 6.1: Summarization of concepts related to UI tailoring.

Considering the presented concepts, the definition of tailoring adopted in this work is the one presented by Mørch [71]. Thus, the focus of this work is to use the WELFIT 2.0 to apply tailoring of level 3 (extension) without the need for the end user's intervention to produce the code. In other words, promote automatic customization or self-adaptive UI, considering structure and layout of evaluated Web pages.

6.2.2 Website evaluation tools

The next subsections present the reviewed tools grouped according to the solution used for gathering usage data, namely: server-side capture, proxy-based capture, and client-side capture.

Server-side capture

Web Utilization Miner is a system for the discovery of navigation patterns in websites, which are represented in digraphs [113]. WebSIFT (Web Site Information Filter) is a system that aims at discovering knowledge from usage, content, and structure of websites [25]. LumberJack is a tool that processes Web server logs and uses the content and hyperlinks to build a model of user activity, applying clustering and a number of statistical analyses to the data [21]. *Descubridor de Conhecimento en la Web* (DCW) is a tool for discovering navigation patterns and association rules in server-side logs [35]. WebCANVAS is an evaluation tool that uses server-side logs. The tool applies clustering techniques to group similar navigation patterns and supports the visualization of these clustered patterns [16].

These WUM tools have provided interesting contributions to the statistical analysis of usage and to the discussion on how to represent the outcomes of an evaluation in a summarized way. However, the logs used in these studies are captured at the server-side and thus do not count on detailed interactions.

Proxy-based capture

WebQuilt is an automatic capture and analysis tool that uses page-view level logs as data source. It uses a proxy-logger that mediates between users and Web servers and stores the communication between them [52]. MouseTrack is a proxy-based usability system that performs automatic client-side capture and analysis. It provides an online configuration and visualization tool that shows the mouse path followed by website visitors [7]. UsaProxy is a proxy-based usability evaluation system that performs automatic capture and analysis of client-side events. It uses JavaScript and focuses on usability tests [8]. WebinSitu is an enhanced version of UsaProxy that focuses on behavior comparisons between blind and sighted users [13]. WAUTER is a proxy-based Web usability tool. It employs a functional set of tools that automate the capture and analysis of usability evaluation methods through the use of client-side logs and task models [9]. Web Usability Probe (WUP) is a proxy-based remote usability evaluation tool that considers formal use. The tool uses client-side logs as data source and the vocabulary of events, which stands for the number of different event types, including all JavaScript events. In addition, it allows the definition of customized events, giving evaluators the flexibility to add specific events to be detected and considered in the evaluation. The tool enables the evaluator to perform a comparison between actual user behavior and an optimal sequence of actions [17].

In some cases, proxy-based tools require reconfiguration of the user's browser or a proxy setup. Another shortcoming involves the integration of code into evaluated Web pages, which may be problematic and cause malfunctioning of the logger module. Moreover, requests performed using technology different than the one used in the capture module is harder or even impossible to capture (e.g., Java Applet or Flash).

Client-side capture

WebVIP is a logger for formal tests. The vocabulary of events is restricted to a few events (i.e., press/hold keys, press/hold/move the mouse pointer, enter/leave a widget, and enter/exit the window). The environment configuration requires a local copy of the entire website being evaluated [77]. WET is another example of a logger for formal tests. It uses cookies to store logged data, leading to the reduction of the vocabulary of events due to storage issues [37]. These loggers represent the first efforts to capture client-side events.

WebRemUSINE is a tool that performs the automatic capture and analysis of website interaction logs in order to detect usability problems through remote evaluation. The analysis of logs is based on the comparison between the paths used by users and the optimum task model configuration. The data transmission is conducted by a Java applet component. The user must select the tasks s/he is performing so that the events captured can be related to the task selected by the user [83]. MultimodalWebRemUSINE is the latest version of the tool that aims to exploit the possibilities opened up by recent technologies to gather a richer set of information regarding user behavior. The tool allows traditional graphical logs to be analyzed together with the logs from webcams and portable eye trackers [85].

Google Analytics is an automatic capture and analysis tool. The default data source used by the tool represents page-views. The tool requires the evaluator to register him/herself and to insert a JavaScript code into the Web pages to be evaluated. It provides different report formats, allows actions to be registered as virtual page-views, and has a feature to register customized events at the client-side. These customized events in Google Analytics are events that can be named by the evaluator and triggered in any Web page component configured to communicate with the JavaScript data-logger (e.g., a Flash video or HTML event handler). However, the tool has a limit of logging 500 customized events per visit [43]. Google Analytics is the most popular evaluation tool, being used in 53.5% of websites [119].

WELFIT is an evaluation tool that captures logs triggered at client-side and analyzes these logs in order to point possible usage incidents [100]. It provides summarization and presentation of usage logs allowing the identification of patterns. The tool depends on the JavaScript support and on the participant acceptance on being part in the evaluation. The WELFIT was used as the technological basis to develop the self tailoring approach presented in this paper.

Although researchers have shown interest in this field and a number of automatic evaluation tools have been proposed, some gaps still remain. A tool can automate the usage capture and analysis, but this kind of tool requires a setup and specific client configuration. In addition, some tools can keep track of users' mouse movements but are unable to capture usage data from the wide diversity of users, independent of the device used to surf the Web [101]. Other techniques used to log or store usage data reduce the event vocabulary or the number of events captured in each session. Additionally, task modeling has been advocated by some authors [85, 83, 113], but the creation and maintenance of the task model can be time-consuming for developers. The tools should avoid requiring significant effort on the part of the participants in the evaluation (e.g., configuration of the browser, access to a bookmarked proxy link), as well as on the part of the evaluator. Finally, according to a survey performed by Ivory and Hearst [59], 3% of evaluation tools automate critic, i.e., present suggestions to evaluators on how to eliminate usage incidents identified in the analysis phase. In this work, adjustment is considered as the next step regarding automation on evaluation tools. Thus, the automation of capture, analysis, and adjustment is an approach, to the best of our knowledge, not presented in the literature and it is explored in this work.

6.3 Theoretical and Methodological References

The main theoretical reference considered in this work is the Organizational Semiotics (OS) [115]. OS is a discipline that deals with information and information systems taking into account technical issues as well as social and human aspects [117]. The OS counts on a set of methods named MEASUR (Methods for Eliciting, Analyzing and Specifying Users' Requirements), which can be used for the understanding, developing, managing, and using information systems [62]. In the context of UI evaluation OS has an important role, since those tools need to consider issues from computational infrastructure to user-centered and evaluator-centered issues, dealing with concepts of accessibility and usability.

The methods of MEASUR applied in this work are the following [62]:

- Problem Articulation Method (PAM), that supports the clarification of the problem;
- Semantic Analysis Method, which supports the elicitation and representation of requirements;
- Norm Analysis Method (NAM), which provides means to specify general patterns of behavior of all agents within the system.

The PAM artifact used was the Semiotic Ladder (SL), which supports the analysis of information systems in six different layers of signs [116]. It contributes with the clarification of what is needed in terms of an information system that solves not only problems related to the information technology platform, but also considers social aspects of system usage. The SL was used in order to organize evaluation tools' characteristics and define requirements for evaluation tools based on event logs. The requirements defined in Santana and Baranauskas [101] were revisited in Santana and Baranauskas [104]. The NAM helped with the identification of patterns and their implementation in a norm general shape:

If CONDITION, then CONSEQUENT

NAM was applied for establishing the semantic model of the self tailoring system proposed, constructed upon the WELFIT 2.0's ontology chart (Figure 6.2), which is part of the Semantic Analysis Method. Thus, once behavioral patterns are identified, a condition for selecting a specific occurrence of these patterns inside event streams is defined (selector). Then, once the pattern on the condition part is found, the consequent part is then applied in the form of adjustment in the UI that the condition refers to. In short, WELFIT 2.0 applies norms of adjustment into the evaluated UI.

The ontology chart (Figure 6.2) represents the following facts: organizations provide websites; Web pages are part of a website and websites are part of the Web; evaluators use evaluation tools to analyze websites; logs are a product of Web page usage; an evaluation tool that analyzes logs can count usage incidents; the generation of adjustments depends on an evaluation; finally, to apply adjustments the tool depends on the generation of them.

The requirements that guided the construction of WELFIT 2.0 presented elsewhere [104] are summarized in the following SL's layers:

- **Social layer** – integration of A&U for the target audience of the evaluated website; enable remote testing during real use of the evaluated website; interfere with the Web

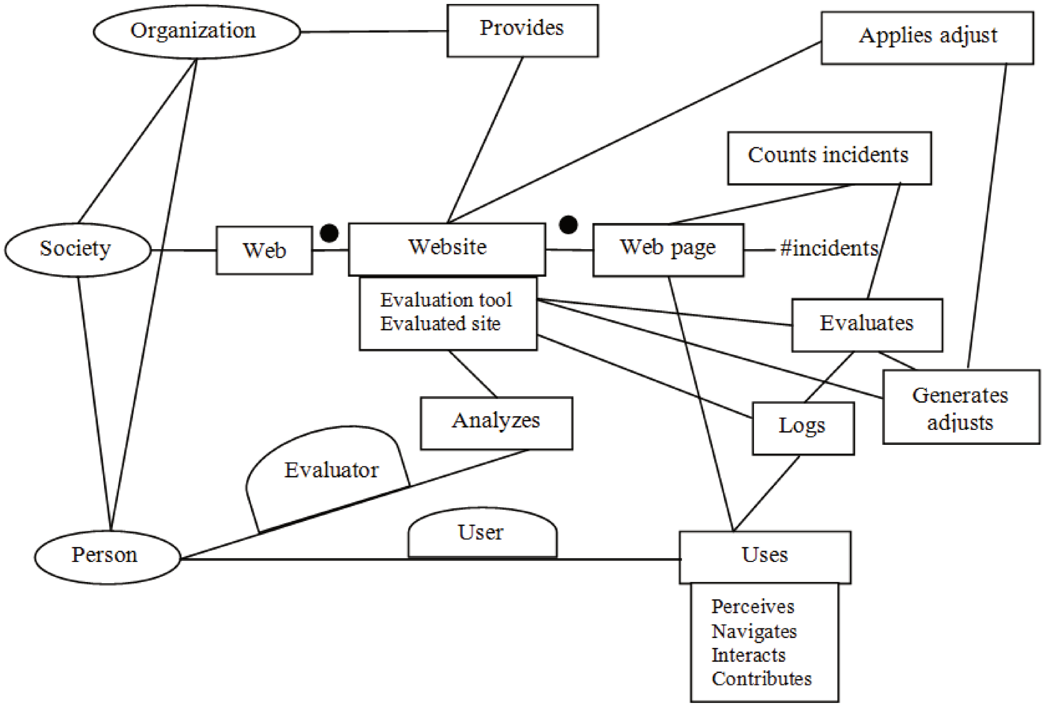


Figure 6.2: Ontology chart of WELFIT 2.0.

page operation as minimum as possible; the evaluation report should call administrator's attention to points where users may have faced problems using the website; the tool's widget should allow the customization of its own appearance in order to be gracefully incorporated into the evaluated website's design.

- **Pragmatic layer** – the tool should require two actions: one to start the capture, which stays valid for future sessions, another to interrupt the capture, which may occur at any time; tool's controls must be gracefully combined with the design of the website being evaluated; the intention passed to the user by the website and controls should result in an integrated experience.

- **Semantic layer** – provide high levels of abstraction without depending on specific task models, grammars, or events; provide controls representing the status of the tool and user context during the test session; the evaluation report should allow administrators to view, in one spot, data concerning visits of one or more users; if the user can not define settings in a certain environment, the answers and other data entered by the user should be recorded and kept by the website being evaluated.

- **Syntactic layer** – use all available data to obtain correlations between them; the combination of the available data in different components may reveal information impossible to obtain independently; use high granularity data to allow the discovery of low and high level patterns.

- **Empirical layer** – prevent that processing or transmitting logs interfere with the use of the evaluated interface; the tool should implement safe and effective techniques to transfer logs without impacting on the website usage; the time required to generate the report must consider practical limits if the intention is to retrieve it synchronously to the administrator.
- **Physical layer** – do not depend on resources or specific configuration of the participants devices (e.g., disk space, bandwidth, etc); the evaluation tool should include mechanisms to achieve their goals in different configurations of hardware and software.

Finally, the presented references guided the development of the WELFIT 2.0 combining the use of observational data to perform different adjustments for each cluster of users in each evaluated Web page, continuously.

6.4 Self Tailorable Approach

This section details the self tailoring approach implemented on the WELFIT 2.0. The new version changed the system from a tool that identifies possible usage incidents and summarizes observational data into an evaluation tool that provides a self tailoring feature aiming at the removal of accessibility barriers and usability problems.

Figure 6.3 presents an overview of the architecture and technologies involved in the development of the WELFIT 2.0. The 2-tier architecture counts on two main modules: server module, which follows a 3-layer architecture inspired by the MVC (Model-View-Controller) architectural pattern and is responsible for offering the administration interface, providing reports, applying data mining techniques, and persisting logged data into the database; client module, which is responsible for inviting users, capturing and compacting events, sending logged data to the server, requesting adjusts to the server, and applying the adjusts.

The client module is implemented as a jQuery plug-in. The jQuery[1] is one of the most used JavaScript libraries, which allows developers an abstraction level dealing with cross browser issues, besides facilitating tasks of manipulating Web page elements' properties. This fact was also important in the rationale of considering this library, since adjustments can be applied as pure JavaScript commands or jQuery-like JavaScript commands.

At the server module the presentation layer counts on Web pages using Java Server Pages (JSP) technology and Java Applets for presenting the graph reports. The controller

[1] http://jquery.com/

layer is responsible for controlling requests by verifying if all the data sent to the server are in the proper format/structure. Then, once all verification is performed at the controller layer, then the services are requested to the core of the WELFIT 2.0, present at the model layer. The model layer counts on all WUM techniques, the domain model classes, the adapter component that communicates with the graph library used for presenting usage graphs, and the proper data access objects that stores/retrieves data at/from the MySQL database.

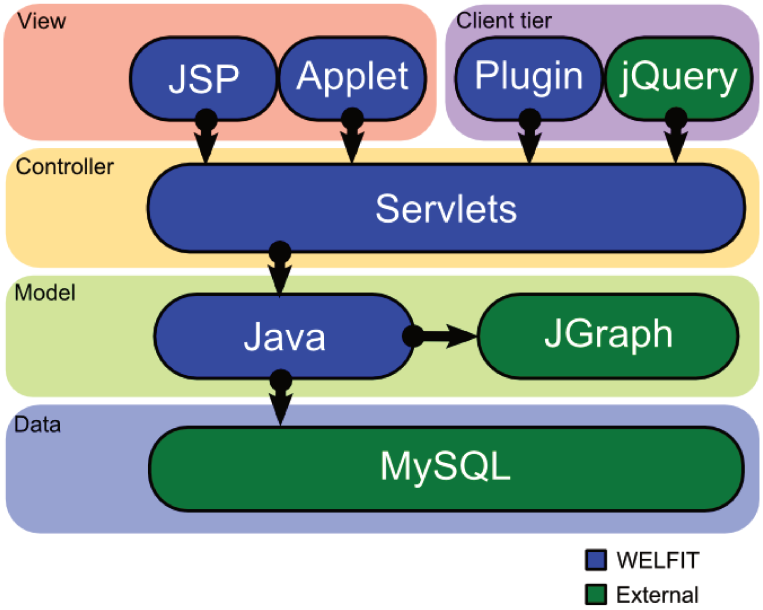


Figure 6.3: Overview of the architecture and technologies involved in the WELFIT 2.0.

6.4.1 Data capture

Based on the previous version of WELFIT, which provided logging and reporting features, the new version dealt with two challenges involving the identification of elements without id attribute and the capture of data while changing between pages. WELFIT 2.0 considered solutions presented in WUP [17], in which ids are generated based on the position of the elements in the Document Object Model Tree and all the pending data are sent before the unload event.

The WELFIT 2.0 logs events triggered at client-side, for example: mouse movements, mouse clicks, key strokes, etc. These events count on information as timestamp, target element, X-Y coordinates, among others. The data capture occurs under users' acceptance in taking part of the evaluation. In order to start the capture the following steps are required:

– The evaluator needs to register her/himself in the administrative interface of the WELFIT 2.0;

– Once authenticated, the evaluator can register the websites s/he wants to evaluate;

– As soon as a website is registered, the evaluator can include the client module call into all Web pages to be evaluated.

When a visitor first accesses a website under evaluation, an invitation is presented. If the user accepts taking part in the evaluation, the capture starts; otherwise the invitation is closed. In both cases the response is recorded in an application cookie preventing users to answer the invitation more than once. Explicitly requiring users to answer the invitation is an ethical requirement, although it may result in less subjects to be observed. The experiment design deals with this issue by dividing each cluster randomly into experimental and control parts. This division will not exclude any participant, since it will be applied only while an adjustment is being experimented. When specific adjustment results in less user incidents for the experimental part, then the adjustment is considered successful and is applied to all participants placed in the same cluster.

Besides the question related to the invitation, the tool's widget also shows the following yes/no question: "Are you using assistive technology?" This information is used to analyze the behavior of users that use assistive technology and users that do not. Moreover, this information is important to support the characterization of accessibility barriers. The invitation is presented in the position defined by the evaluator. The WELFIT 2.0 also allows evaluators to define specific CSS (Cascading Style Sheet) for stylizing it gracefully with the evaluated Web pages, since this is relevant for users to participate in the evaluation [104].

6.4.2 Identifying Usage Incidents

The reports provided by the tool summarize the logged data and support the evaluators in identifying patterns in the UI usage. The reporting feature allows evaluators to verify different association rules and to check most repeated event sequences. In addition, the reports count on a heuristic to point out potential usage incidents. The heuristic is based on a metric that uses the fact that the sequence of events triggered up to a certain element allows to identify significant differences among event streams without depending on task models or specific grammars [100].

One possible analogy to this metric, in the context of architectural accessibility, is to consider two persons that are in a point A and want to go to a point B (Figure 6.4). Some people can walk through the straight path from A to B (Figure 6.4, left). However, part of the population may need a different path to go from point A to point B, resulting in a longer path (Figure 6.4, right). Hence, the metric indicates different behaviors so that the comparison considering the different ways of reaching point B might indicate a usage incident. In this example the accessibility barrier can be considered the lack of support to all people to use the shortest path from A to B, e.g., a person using wheelchair or a blind person using a cane. Thus, the metric is a way of measuring distances among behaviors, represented by event streams.

Usually WUM studies that require a distance measurement consider Euclidean distance as a first approach. However, when using Euclidean distance to define distance between event streams, structural information regarding the order in which the events occurred is not considered. Aiming at considering structural information contained in the event streams, our approach is based on the Sequence Alignment Method (SAM) [49]. SAM, also named Edit Distance, is a measure that represents the amount of operations required to transform a source sequence ($x[1,...,m]$) into a target sequence ($y[1,...,n]$) [49, 45]. Each operation has a weight. The operations considered for transforming the source sequence into the target sequence are: copy (C), replacement (R), deletion (D), insertion (I), and twiddle (T) [26]. Combining the presented operations and their weights the SAM distance (d_{SAM}) between event streams S_1 and S_2 is calculated according to Equation 6.1.

$$d_{SAM}(S_1, S_2) = w_d D + w_i I + w_r R + w_t T \tag{6.1}$$

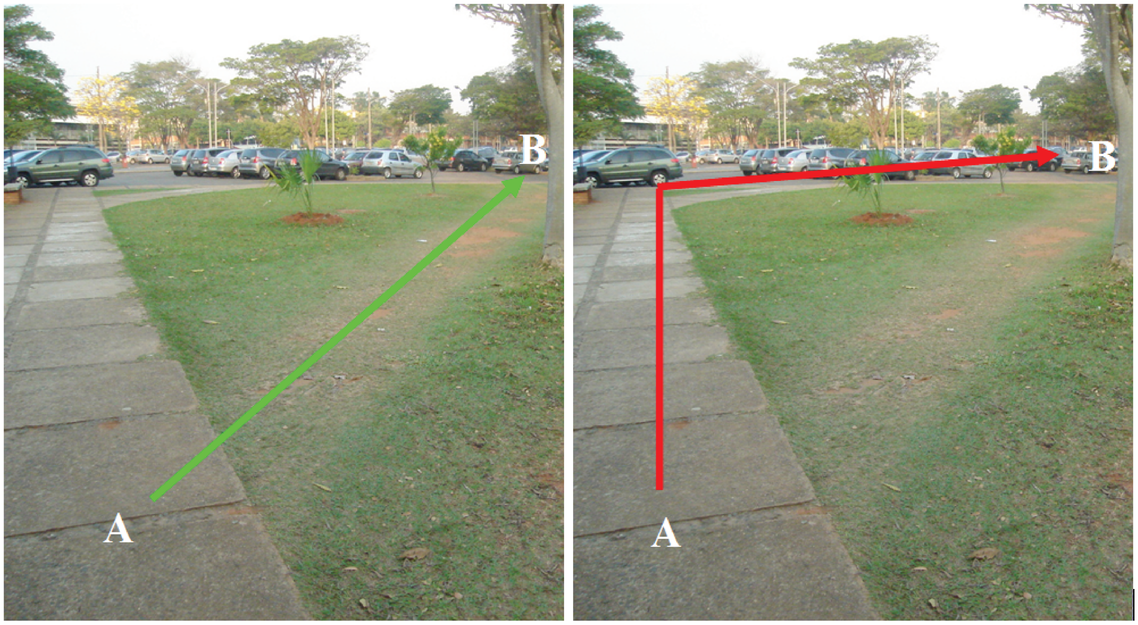


Figure 6.4: Example of the metric considered in the heuristic.

The heuristic is based on SAM and on graph properties. The graph properties are applied to a usage graph, a directed graph where each vertex represents a certain event on a certain UI element and the edges reflect the order in which they had occurred. In addition, vertices of the usage graphs have information as the number of sessions in which it occurred, and the mean distance to the root node (i.e., the start of the UI usage) [100].

In short, the heuristic aims at using the mean distance to a certain node and the way this measure changes across the flows in the outgoing nodes. Thus, from a certain node u, if there is an outgoing node (v) with a significant higher mean distance, then there is an alternative walk to v, indicating that it might be part of a usage incident, just as it would occur counting steps performed from A to B in Figure 6.4. More details regarding the SAM-based heuristic can be found in [105].

Finally, the SAM-based heuristic supports the task of counting possible usage incidents, which is used for analyzing results of adjustments; d_{SAM} is also considered in this work for defining clusters of event stream. This last use of SAM will be detailed in the next subsection.

6.4.3 Event Streams Clustering

From the d_{SAM} distance calculated between all pairs of event streams for each Web page, WELFIT 2.0 considers an agglomerative hierarchical approach to obtain clusters. The agglomerative hierarchical approach considers initially that each element is an individual cluster and then the closest elements are grouped. Thus, a proximity function and a final fusion function are needed [42]. In addition, the Average Linkage algorithm has shown to be more adequate to the data considered since it avoids negative points of Single Linkage and Complete Linkage as being less susceptible to noise and avoiding specific formats of clusters [120].

The number of clusters (K) is defined in running time considering the sum of the cohesion of all clusters of a certain Web page. The cohesion is defined according to the Equation 6.2.

$$\begin{aligned} Cohesion(Ci) = \sum_{(S_i,S_j)\in C_i} d_{SAM}(S_i,S_j) \\ C_i : \text{Cluster of a Web page} \\ d_{SAM} : \text{Distance between two different event streams} \\ S_i : \text{Event stream of } C_i \\ S_j : \text{Event stream of } C_i \end{aligned} \tag{6.2}$$

Thus, as event streams are clustered, while the Average Linkage algorithm runs, the total cohesion is calculated and the growing of the total cohesion is analyzed at each step.

Hence, when a significant growing occurs, it means that clusters with adequate cohesion are being merged, then the clustering algorithm stops [120].

6.4.4 Norms Specification

Norms reflect regularities in the behavior of members in an organization [114]. Thus, in the WELFIT 2.0 the feature offered regarding adjustments is based on the specification of norms. In the system, norms have a description, an event stream selector, and an adjustment (that might refer to selected UI elements) to be applied into the evaluated UI.

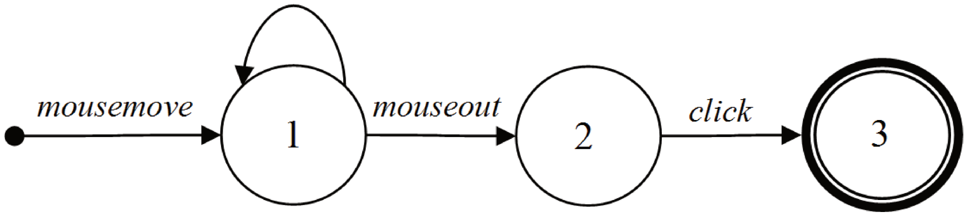
The selector component plays an important role on the norms considered in the tool. The selector follows a regular expression format concatenating event name and an id-based name to refer to the UI element (e.g., *click@logo*). The event can be any of the standard JavaScript or jQuery events captured by the tool. The id-based name gives the evaluator the possibility of mapping any number of concrete UI elements. This feature does not require the evaluator to know every element's id and allows the same norm to be applied to any Web page. The evaluator can refer to elements as any id-based name and then refer back to the same value given in the adjustment part of the norm. Moreover, the selector may contain indicators of repetitions such as '+' and '*'. In the system this feature is implemented as a state machine that follows the Visitor Gang of Four Design Pattern [41]. The state machine counts on a hash table for keeping the id-based names as references for proper UI elements ids.

For example, the selector part of a norm for matching event streams containing repeated mouse movements over an element and a misguided click in the element containing the clickable element could be defined as:

mousemove@inner+ mouseout@inner click@container

Complementarily, the state machine structure that is created from the definition of the selector above can be represented by the Figure 6.5. Note that the selected UI elements are a result of the state machine functioning. As the state machine processes event streams it fills the proper values of concrete UI elements and, if it reaches the final state, the id-based name and respective concrete UI elements are returned to be registered as a concrete adjustment.

In the given example, the adjustment can use the id-based names inner and container into the JavaScript commands. The evaluators can define adjusting commands using jQuery-like JavaScript commands, facilitating the task of manipulating Web page elements. Hence, for the previous example of selector, a possible adjustment of increasing the size of the inner element could be:



Key	Value
inner	
container	

Figure 6.5: Example of a state machine built from the previously presented selector.

```
jQuery( ‘‘inner’’ ).css( ‘‘width’’,
     Math.round( jQuery( ‘‘inner’’ ).width() * 1.2 ) + ‘‘px’’ );
```

This code retrieves the width of the selected element, increases it by 20%, and then uses the *Math.round*() function in order to consider the rounded value. Note that in the selector considered in this example the inner will be selected for each Web page where the selector will be applied. For example, in a certain Web page the “inner” can be replaced by “#logo” and in other Web page by “#logout”. In addition, for elements that do not count on the id attribute, a solution similar to the one presented in [17] for generating ids is considered. The solution involves generating ids based on the position of the Web page element in the Document Object Model Tree. It was inspired by XPath, as if the id was a selector for the UI element where an event is triggered. Note that this feature supports identifying part of the structure used in the Web page and, consequently, enables tailoring applications to manipulate the part of the UI that is the target of the adjustment.

6.4.5 Adjustments Maintenance

Once presented how WELFIT 2.0 captures usage data, identifies and counts usage incidents, clusters event streams for each URL, and provides the feature of registering norms into the evaluation tool, it is time to present how the tool gathers all these steps and, following the WUM process, creates experiments in order to retrieve adjustments to participants. WELFIT 2.0 approach on tailoring is performed by continuously experimenting adjustments contained in the norms registered in the system.

The tool maintains the distance between every two different event streams for each URL recorded into the database. Thus, this distance is considered for keeping all event streams organized in clusters. In addition, clusters are verified (and updated if it is the case) hourly. This means that the tool computes, for all new sessions, the new distances and places them in the appropriate cluster. Hence, having clusters up to date and knowing that clusters contain correlated event streams, the selection of all registered norms are applied to the most recent event stream per cluster. The most recent stream is considered because the URL may have changed during the experiment and, then the most recent session is the most probable to contain event streams addressing up to date UI elements.

The creation of experiments is performed weekly. The rationale behind this decision is based on the Web 2.0 ‘perpetual beta’ deploys performed weekly, since it considers adjustments as small improvements to be deployed. WELFIT 2.0 performs as follows:

– The WELFIT 2.0 verifies if there is any cluster that fits into any selector registered, for all the defined clusters and all the registered norms.

– If a norm matches an event stream, then the concrete elements found by the state machine are placed into the database and an experiment is created; at this point the experiment has the status of ‘under elaboration’.

– Once a cluster is selected during the elaboration of the experiment, all the event streams that are part of it are read from the database, the usage graph is built, and the total number of usage incidents is computed. At this point the tool counts on the following information: concrete adjustments that resulted from the state machine functioning, URL where the adjustment will be applied, cluster identifier, number of event streams considered, norm applied, number of usage incidents, and sizes of experimental and control parts of the cluster. All this data are placed in the proper register at the experiments database table; at this point the status of the experiment is changed to ‘running’.

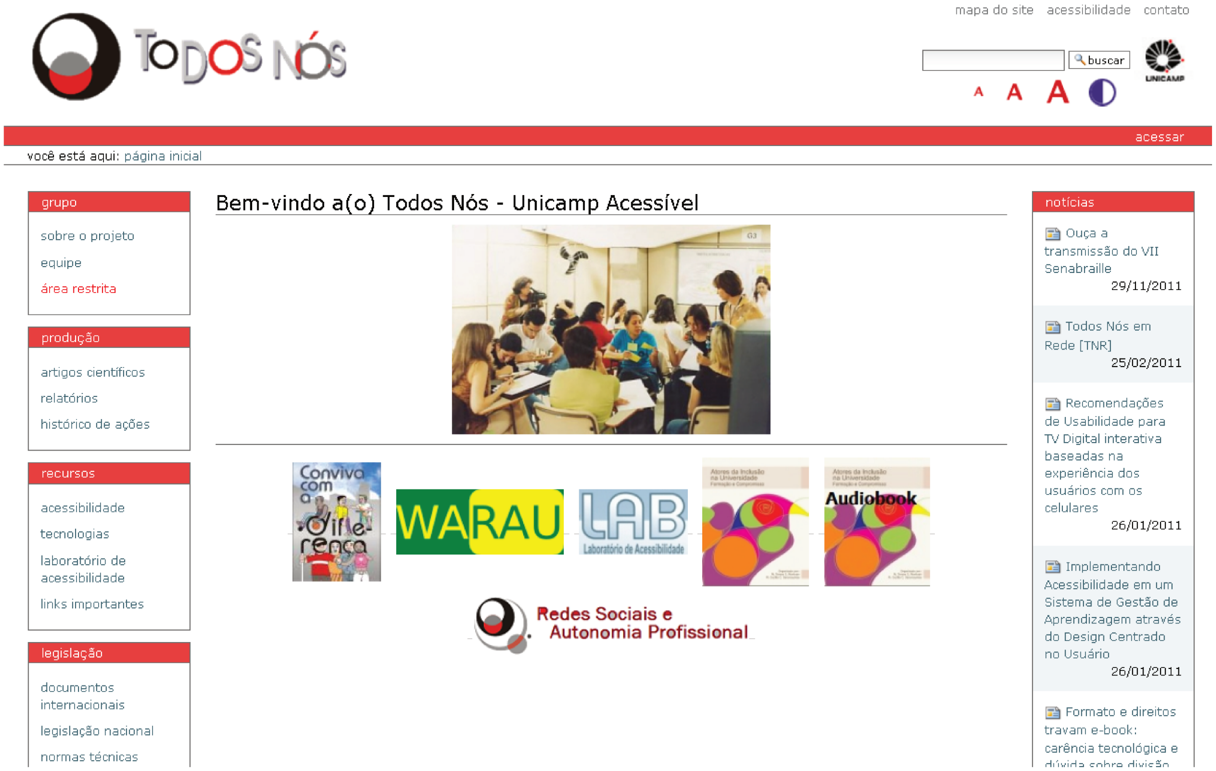
When a participant accesses a URL of an evaluated website, the client module sends the identifier recorded in cookie to the server. The server then verifies if there is a running experiment for that identifier. Thus, if it is the case, the concrete adjustment of the experiment is returned to the user and then applied to the Web page by the client module.

Finally, once an experiment is older than a configurable period (i.e., one week) its success is verified. This task is done by comparing the number of usage incidents of returning visitors. So, for each cluster referred to by a running experiment, the usage incidents of the experimental and control parts are calculated. In addition, if the usage incidents in the experimental part are lesser than in the control part, then the experiment status is changed to ‘successful’ and, in the next visits, the control part of the cluster will also receive the successful adjustment. In other cases, the experiment’s status is changed to ‘unsuccessful’ and the experiment will never be applied to the same cluster.

6.5 Experiment Design and Analysis

The WELFIT 2.0 evaluation counted on 408 participants who were visiting the studied website and accepted the invitation of being part of the evaluation.

Materials: The data set used was captured by WELFIT 2.0 during a 30 months period of logging the interaction of users with the *Todos Nós* website (Figure 6.6). The *Todos Nós* website is the portal of a homonym research group that has the goal of supporting inclusive environments in the university and making possible the access, permanence, and prosecution of college-level education for people with disability. The website includes in its audience people who use AT.



Bem-vindo a(o) Todos Nós - Unicamp Acessível

Ouça a transmissão do VII Senabraille
29/11/2011

Todos Nós em Rede [TNR]
25/02/2011

Recomendações de Usabilidade para TV Digital interativa baseadas na experiência dos usuários com os celulares
26/01/2011

Implementando Acessibilidade em um Sistema de Gestão de Aprendizagem através do Design Centrado no Usuário
26/01/2011

Formato e direitos travam e-book: carência tecnológica e dúvida sobre divisão

Figure 6.6: Homepage of the studied website.

From the period of 30 months considered, during the first 24 months the data was gathered and patterns were studied. During the last 6 months the adjustments were applied and then analyzed. The patterns identified were gathered in previous works [104, 105, 107] and are summarized as follows:

I) Repetitive clicks occur next to small clickable UI elements;

II) Repetitive mouse movements occur over text input boxes that have poor contrast between background and foreground colors;

III) Repetitive mouse movements are performed over clickable elements that do not afford clicking;

IV) Repetitive clicks on non-clickable elements;

V) Repetitive selection of text that have poor contrast between text and background colors;

VI) Repetitive clicks on the same element caused by lack of feedback. The data was captured remotely and during informal use (i.e., participants were not asked to accomplish a predefined task).

Design: The variables considered are the number of usage incidents found, the identification of patterns in event stream considering the norms registered in the WELFIT 2.0, and applying the adjustments in order to check if the number of usage incidents is reduced after the adjustments.

The functioning of WELFIT 2.0 involves computing, for each cluster of event streams occurred in a certain Web page, the number of usage incidents, and verifying norms that apply to these clusters. Then, for each cluster, the adjustments pointed out by the norms are flagged to be applied on half (plus/minus 1 for clusters with even number of event streams) of these participants on next visits. Thus, the considered experiment design is two-groups/pre-test/post-test:

$$\begin{array}{llll} N_e & O_1 & X & O_2 \\ N_c & O_3 & & O_4 \end{array}$$

The N_e represents the experimental part of the cluster that will receive the UI with the adjustments pointed by the norms registered in the tool and the N_c is the control part of the cluster, which continues using the UI without adjustments. In addition, O_1 and O_3 represent observations made by the tool before the adjustment. O_2 represents an observation performed in an adjusted UI after the intervention represented by X. O_4 represents an observation performed in a non adjusted UI. For example, first consider a cluster of participants related to a certain Web page containing the O_1 and O_3 observations. Now consider that a pattern was found in these observations, e.g., the cyclic pattern of mouse over and mouse out movements were applied over a small clickable UI element. In this point the O_1 and O_3 count on a respective total of usage incidents. Moreover, the selector of the norm that refers to the amplification of small clickable UI element would match the event streams related to O_1 and O_3. Hence, once the norm's selector matches an event stream, an experiment for the adjustment is created by WELFIT 2.0 in order to be applied to all participants from N_e in the next observations, namely O_2. Finally, the next observations for Nc will not count on the adjustment, so that the variation related N_e and N_c can be then computed.

Considering that each observation might contain zero or more usage incidents, an adjustment is considered to be well succeeded if the reduction of usage incidents from O_1 to O_2 is lesser than the reduction of usage incidents from O_3 to O_4.

From the presented experiment design, the WELFIT 2.0 has a focus on quantitative data, since usage logs from events triggered at client-side are determinant to prioritize adjustments and apply quantitative comparisons. Moreover, adjustments registered are converted into JavaScript commands that, in turn, are requested by client module in order to adjust evaluated Web pages.

The resultant analysis considers significant differences regarding comparisons of usage incidents between O_1 and O_2 and between O_3 and O_4. Note in the Equation 6.3 that the total of usage incidents in the cluster is weighted by the number of event streams in each cluster. The rationale for doing this is that the number of event streams related to the computation of usage incidents after the application of the adjustment may vary. Thus, if the weighted number of usage incidents is reduced, then the adjustment is considered successful.

$$\frac{incidents(O_2)}{|O_2|} - \frac{incidents(O_1)}{|O1|} < \frac{incidents(O_4)}{|O_4|} - \frac{incidents(O_3)}{|O_3|} \tag{6.3}$$

Once an adjustment is considered successful, it is applied to the whole cluster in the visits occurring after O_2 and O_4.

Threats: Testing and maturation threats could exist if O_3 has the same observation as O_4, since the first usage of a UI can alert users about errors or prepare a user for a second usage. Hence, aiming at neutralizing these threats, the effect of the intervention is compared to a second observation of N_c, guaranteeing that O_3 differs and precedes O_4.

The selection threat occurs in the acceptance of the invitation (auto-selection) because users that are willing to be part in the evaluation may want to obtain evaluation results and their behavior can be related to the will and enthusiasm in participate. However, this threat is related to the rationale of prioritizing participants' privacy. On the other hand, the random division of each cluster into N_e and N_c deals with this threat.

Threats of history, selective mortality, contamination, competitive behavior, compensatory behavior, regression to the mean, and expectation of the experimenter are neutralized because the participants are randomly set as being part of N_e or N_c; participants do not know whether they are part of N_e or N_c, and evaluators do not have contact with participants since the data capture is performed remotely. In addition, the probability of users from N_e or N_c never use the evaluated UI in such a way that their usage will not result in O_2 or O_4, respectively, is the same.

All users that first access a Web page being evaluated, through a JavaScript enabled Web browser, will receive the invitation, which is a way of dealing with the threat of coverage, since JavaScript is supported by all modern browsers and is used in 96% of most popular homepages [131].

A threat that may interfere is related to the tool's effect in the participant; users that search for more information about the tool may avoid exploring an evaluated website freely since they know that usage data are being captured and will indicate their behavior. Thus, users may be more careful before triggering any action, which may result in less triggered events. Moreover, participants may explore the system faking actions with the intent of seeing the tool's results.

Finally, the users are identified by values recorded in cookies present in the users' devices. Thus, if a participant's computer is used by a different person, the instrumentation threat may occur. This means that, in a certain moment, a device can contain a cookie identified as being part of N_e or N_c but in fact the computer is being used by a person different from the one that accepted being part in the evaluation. In this work there is no way of mitigating this threat, since the sharing of user accounts in an operational system brings this threat to UI remote evaluation.

6.5.1 Results

During the last 30 month period in which the *Todos Nós* website was evaluated, the tool identified 408 different cookie values registered into participants' devices. The total of logged sessions was 584; 176 referring to returning participants.

The clustering of event streams considering their cohesion is a key step in the approach presented, since adjustments are selected and applied considering the clusters disposition. Figure 6.7 presents how the cohesion value increases at each step of the Average Linkage algorithm at the *Todos Nós* homepage, which is the most important page of the studied website. In this example the homepage counted with an initial number of 55 individual clusters (i.e., 55 sessions or K_0=55) and, after 23 steps of the algorithm in agglomerating closest clusters, it stopped considering how the cohesion grows. In this example the final number of clusters was 32 (K=32). Figure 6.7 also shows that the cohesion increases in a cubic rate, as the proximity function (αX^3, for $\alpha = 2$) presented follows the shape of the calculated cohesion. This result supports future analysis of the final fusion function in order to improve the cohesion of the final K clusters. In addition, it characterizes how the cohesion measure increases when dealing with event streams captured during the usage of a website.

Regarding the functioning of the Average Linkage, we obtained the following results: mean K_0=4.99 (SD=9.02), mean K=2.65 (SD=5.80), and mean number of steps agglomerating clusters of 2.34 (4.65). These results give an overview of the data considered in this work, since the initial number of individual clusters is up to 14 event streams in the

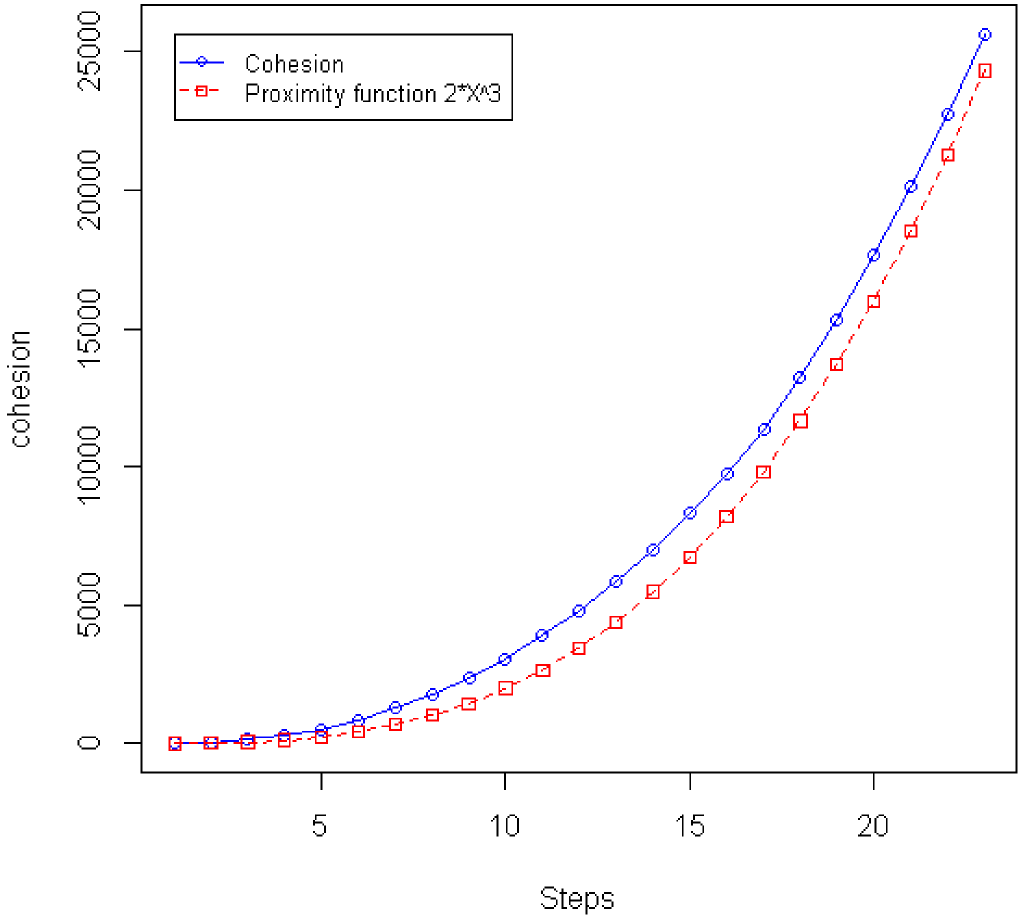


Figure 6.7: Cohesion value vs. steps performed by the Average Linkage clustering algorithm.

majority of cases. In addition, the final number of clusters is up to 8 in the majority of the cases.

The patterns identified in [104, 105, 107] and considered in this work were converted into the following selector-adjustments pairs:

(I) Increase the size of small clickable elements.
Selector: mousemove@outer* mouseover@inner mousemove@inner*.
Adjustment: welfit.jQuery(“inner”).css(“width”, Math.round(welfit.jQuery(“inner”).width() * 1.2) + “px”) ;

(II) Increase the contrast between background and foreground colors when there is repetitive mouse movements over text input boxes.

Selector: mouseover@input mousemove@input+ click@input.
Adjustment: welfit.jQuery(“input”).css(“background-color”, “#efefef”) ;

(III) Highlight clickable elements.
Selector: click@outer+ mousemove@outer+ mouseover@inner mousemove@inner+ click@inner.
Adjustment: welfit.jQuery(“inner”).css(“border-bottom”, “0.2em solid inherit”) ;

(IV) Clicks performed on non-clickable elements.
Selector: mousedown@element blur@element click@element.
Adjustment: welfit.jQuery(“element”).css(“cursor”, “text”) ;

(V) Change background of elements with text for people that select the text while reading.
Selector: mousedown@text mousemove@text+ mouseup@text.
Adjustment: welfit.jQuery(“text”).css(“background-color”, “#ffffef”);

(VI) Repetitive clicks on the same element caused by lack of feedback.
Selector: mousedown@element mousemove@element* mousedown@element.
Adjustment: welfit.jQuery(“element”).css(“cursor”, “text”) ;

The adjustments are applied not only for the URL it refers to but to the whole website. Then, all adjustments referred to a certain participant are forwarded to the client module. Hence, the client module has the responsibility of verifying if the component that each adjustment refers to is present in the Web page. This also increases consistency across Web pages if an adjusted UI component appears in pages other than the one that the participant had used.

The adjustments were available for participants during the last 6 months of the 30 months period considered in this work. From the presented norms and considering all logged sessions, a total of 110 experiments where created by the tool. Considering the analysis performed as presented in Equation 6.3, 19 experiments were well succeeded. The most successful norms were I, IV (Figure 6.8), and V (Figure 6.9). Respectively they obtained 10, 4, and 2 successful results.

6.6 Conclusion

The problem addressed in this paper presupposes that different user needs may emerge in different and not foreseen usage contexts; moreover, inclusive solutions regarding website interfaces might be constructed from their continuous evaluation based on users’ behavior.

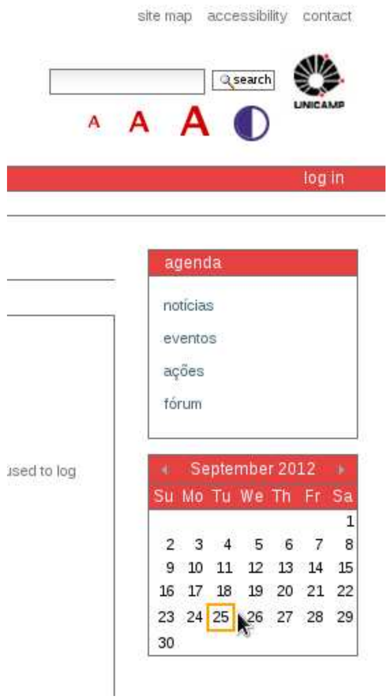


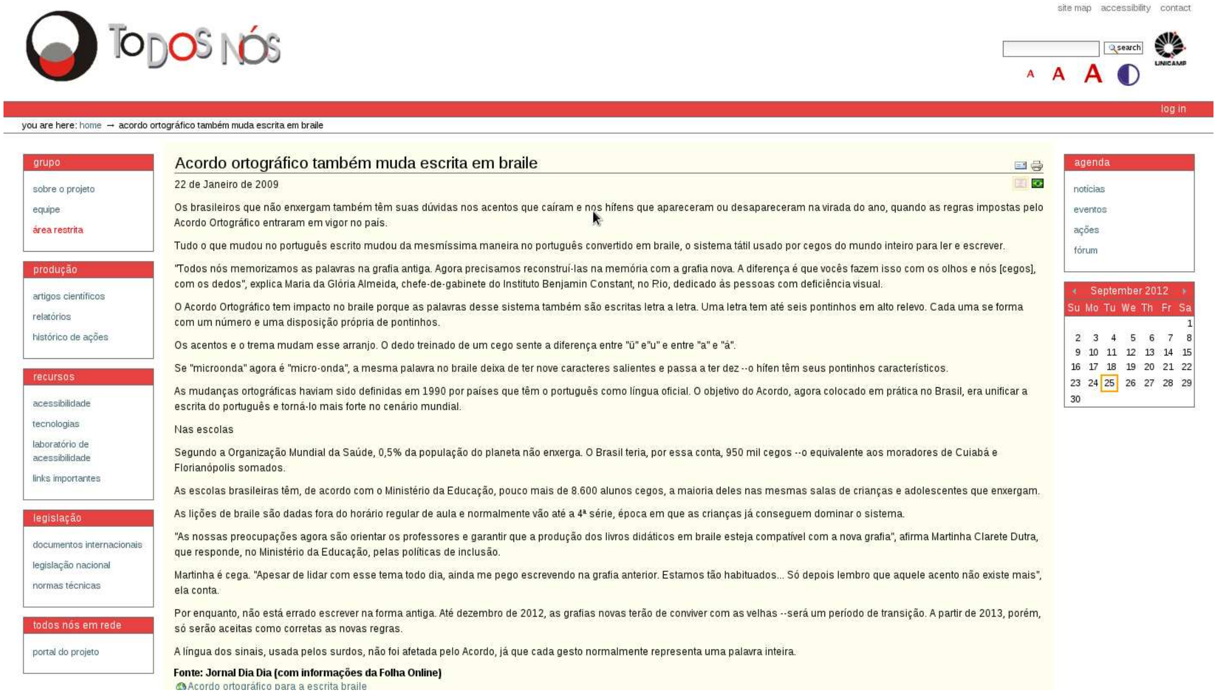
Figure 6.8: Example showing the calendar feature that the WELFIT 2.0, according to the norm IV, identified misguided clicks.

This work presented the WELFIT 2.0, an evaluation tool that offers a self tailoring feature. The approach involved applying pieces of JavaScript code in order to adjust the UI, according to users' behavior. Adjustments considered ordering, size, and color of UI elements.

The use of id-based names in the adjustments allows the pieces of code to be reused in different Web pages, without depending on task models. In fact, one norm has the power of being applied in any page of any website, requiring only one occurrence of the event stream that matches the norm's selector.

The obtained results revealed that the tailoring of level 3 (extension) is a promising way of adapting Web UIs considering the users' behavior. Moreover, the approach revealed that issues related to different user needs may emerge during usage and that the self tailoring approach can deal with these needs by analyzing the real use. In the presented approach the continuous evaluation focuses on eliminating accessibility barriers and usability problems, supporting the Design for All.

In relation to possible limitations of this work, the website considered does not have the pretention of representing the whole diversity of the Web. However, it supported the validation of the proposed approach on self tailoring UI by increasing elements' size, increasing elements' contrast, and repositioning elements for people that use AT. Finally,



you are here: home → acordo ortográfico também muda escrita em braile

Acordo ortográfico também muda escrita em braile

22 de Janeiro de 2009

Os brasileiros que não enxergam também têm suas dúvidas nos acentos que caíram e nos hífens que apareceram ou desapareceram na virada do ano, quando as regras impostas pelo Acordo Ortográfico entraram em vigor no país.

Tudo o que mudou no português escrito mudou da mesmíssima maneira no português convertido em braile, o sistema tátil usado por cegos do mundo inteiro para ler e escrever.

"Todos nós memorizamos as palavras na grafia antiga. Agora precisamos reconstruí-las na memória com a grafia nova. A diferença é que vocês fazem isso com os olhos e nós [cegos], com os dedos", explica Maria da Glória Almeida, chefe-de-gabinete do Instituto Benjamin Constant, no Rio, dedicado às pessoas com deficiência visual.

O Acordo Ortográfico tem impacto no braile porque as palavras desse sistema também são escritas letra a letra. Uma letra tem até seis pontinhos em alto relevo. Cada uma se forma com um número e uma disposição própria de pontinhos.

Os acentos e o trema mudam esse arranjo. O dedo treinado de um cego sente a diferença entre "ü" e"u" e entre "a" e "â".

Se "microonda" agora é "micro-onda", a mesma palavra no braile deixa de ter nove caracteres salientes e passa a ter dez --o hífen têm seus pontinhos característicos.

As mudanças ortográficas haviam sido definidas em 1990 por países que têm o português como língua oficial. O objetivo do Acordo, agora colocado em prática no Brasil, era unificar a escrita do português e torná-lo mais forte no cenário mundial.

Nas escolas

Segundo a Organização Mundial da Saúde, 0,5% da população do planeta não enxerga. O Brasil teria, por essa conta, 950 mil cegos --o equivalente aos moradores de Cuiabá e Florianópolis somados.

As escolas brasileiras têm, de acordo com o Ministério da Educação, pouco mais de 8.600 alunos cegos, a maioria deles nas mesmas salas de crianças e adolescentes que enxergam.

As lições de braile são dadas fora do horário regular de aula e normalmente vão até a 4ª série, época em que as crianças já conseguem dominar o sistema.

"As nossas preocupações agora são orientar os professores e garantir que a produção dos livros didáticos em braile esteja compatível com a nova grafia", afirma Martinha Clarete Dutra, que responde, no Ministério da Educação, pelas políticas de inclusão.

Martinha é cega. "Apesar de lidar com esse tema todo dia, ainda me pego escrevendo na grafia anterior. Estamos tão habituados... Só depois lembro que aquele acento não existe mais", ela conta.

Por enquanto, não está errado escrever na forma antiga. Até dezembro de 2012, as grafias novas terão de conviver com as velhas --será um período de transição. A partir de 2013, porém, só serão aceitas como corretas as novas regras.

A língua dos sinais, usada pelos surdos, não foi afetada pelo Acordo, já que cada gesto normalmente representa uma palavra inteira.

Fonte: Jornal Dia Dia (com informações da Folha Online)

Acordo ortográfico para a escrita braile

◂ September 2012 ▸

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Figure 6.9: Adjusted UI after the application of the adjustment generated from the norm V.

this work considered adjustments regarding structure and layout. No content adjustment was considered, since natural language would be required in order to manipulate content information and this is out of the scope considered in this work.

Further work involves: to extend the set of websites where to apply the proposed approach and to improve the state machine considered in the norms' selectors to provide more powerful selectors (e.g., considering a certain number of occurrences of a UI event or even considering additional information as access keys in the selectors).

6.7 Acknowledgments

The authors would like to thank *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP) for supporting this research (grant #2009/10186-9) and all participants. In addition, we also thank JGraph.com for providing one academic license of the JGraph Layout library.

Chapter 7

Conclusão

O conceito de *tailoring* está sendo reconhecido, cada vez mais, como um valor em sistemas interativos, possibilitando que pessoas tenham cada vez mais produtos/serviços mais adequados às suas necessidades. *Tailoring* pode ser aplicado em diferentes níveis. Os níveis 1 e 2, i.e., customização e integração, respectivamente, já são utilizados em diversos sistemas de informação, e.g., Microsoft Word® e Adobe Photoshop®. Na Web estes dois níveis já são apresentados em alguns *websites* e já são comuns em alguns gerenciadores de conteúdo (e.g., Drupal e Plone). O nível 3 de *tailoring* é considerado em poucas aplicações, pois requer maior nível de conhecimento por parte da pessoa que insere o código novo. O gerenciador de conteúdo Drupal é um exemplo de sistema que permite a inserção de código em *websites*. Considerando a Web como um todo, sem exigir conhecimento específico para desenvolver código, o nível 3 de *tailoring* é um problema ainda em aberto. E é nesta direção que esta tese estende o estado da arte, possibilitando que diferentes *websites* apliquem *tailoring* de nível 3 sem exigir que os usuários finais gerem código.

Após as diversas etapas que fizeram parte desta tese foi possível concluir que a representação visual proposta (grafo de uso) permite a visualização das cadeias de eventos relacionadas às sessões em uma determinada página Web, representando transições comuns e revelando padrões de uso, destacando *outliers* tendo como base a heurística apresentada no capítulo 2. Complementarmente, a captura utilizada no WELFIT 2.0 não interferiu na maneira como participantes interagem com o *website* em avaliação. Ainda, a representação das cadeias de evento para potenciais avaliadores foi corretamente interpretada pela maioria dos participantes.

Este trabalho partiu do pressuposto que avaliar código fonte de páginas Web não é suficiente para desenvolver *websites* acessíveis e usáveis. E o uso de *logs* obtidos no lado do servidor não permite que avaliadores obtenham informações detalhadas sobre como um *website* é usado. Por outro lado, *logs* coletados no lado do cliente fornecem informações detalhadas em relação a quais ações foram executadas e quais elementos de IU foram usados. A abordagem apresentada envolve a inserção de trechos de código JavaScript com o intuito de ajustar a IU, considerando o comportamento dos usuários. Os ajustes consideraram estrutura e *layout*. A proposta considerando identificação dos elementos de IU utilizados nas cadeias de eventos possibilita que trechos de código sejam reutilizados em diferentes páginas Web, sem depender de modelos de tarefa. Complementarmente, uma norma, contendo um ajuste abstrato (i.e., que não referencia elementos concretos de IU), pode ser aplicada em qualquer página de qualquer *website*, exigindo apenas que uma ocorrência do seletor da norma seja identificada na cadeia de eventos.

Em suma, a solução apresentada nesta tese considera dados da utilização real de *websites*, utiliza a heurística para identificar incidentes de uso, se baseia em normas para identificar padrões em cadeias de eventos, a partir desses padrões relaciona os elementos de IU referenciados nas cadeias de eventos, cria experimentos para os ajustes cadastrados nas normas e incorpora código novo nas páginas Web avaliadas.

Os resultados obtidos na tese indicam que: a heurística apresentada aponta incidentes de uso com precisão de cerca de 65%; a abordagem proposta para captura e transmissão de dados de uso capturados no lado do cliente não interfere na utilização de IU avaliada quando considerada em um contexto de baixo letramento digital; a apresentação de relatórios via grafo de uso é interpretado corretamente pelo público alvo em mais de 60% dos casos; o WELFIT 2.0 auxilia na identificação de padrões de uso, assim como na identificação de diferenças significativas na composição das cadeias de eventos relacionadas a participantes que usam TA e não usuários de TA; a nova abordagem apresentada para ajustar IU automaticamente é viável e pode ser aplicada em estudos para melhorar estrutura e *layout* de IU que são utilizadas por pessoas com e sem deficiência, representando assim a principal contribuição desta tese em direção ao *Design* para Todos.

A utilização de eventos do lado do cliente como fonte de dados faz com que a solução proposta seja robusta em relação ao surgimento de novas tecnologias. Dessa forma, mesmo considerando novos dispositivos de interação (e.g., telas de toque, acelerômetro e câmeras que detectam movimento), é possível analisar os eventos resultantes da interação dos usuários com esses dispositivos.

A abordagem apresentada revelou que questões relacionadas a diferentes necessidades de usuários potencialmente surgem, fazendo com que a proposta de *tailoring* apresentada possa lidar com essas questões através da análise do uso real; aspecto não endereçado durante o *design*, projeto ou desenvolvimento. Esta tese apresentou uma abordagem que estende o estado da arte apresentando uma abordagem inclusiva que considera o uso. Por fim, trazer usuários reais para avaliações de *websites* é uma forma de obter dados mais representativos e, consequentemente, contribuir para que a IU seja mais inclusiva.

7.1 Discussão

Uma das perguntas iniciais do trabalho indagava sobre relação da captura de dados em um domínio específico (e.g., dispositivos móveis) para auxiliar no ajuste de uma IU em outro domínio (e.g., *desktop*). Entretanto, na literatura relacionada a dispositivos móveis e no estudo de como contextos de uso diferentes impactam significativamente na forma como uma IU é utilizada, há resultados que indicam que o uso é significativamente diferente, o que nos leva à conjectura de que um ajuste aplicado para uma IU em um determinado domínio com base em um padrão de uso identificado poderia tanto reduzir quando causar incidentes de uso em outro, dado que contam com variáveis significativamente diferentes.

No início do trabalho também foram levantadas questões relacionadas ao número de ajustes aplicados. A principal delas está relacionada ao fato de que se dois ou mais ajustes são identificados para um determinado grupo de usuários, deveriam eles ser aplicados todos em conjunto ou um de cada vez? Esta questão levanta outras duas:

1) Qual ajuste aplicar primeiro, caso os ajustes sejam serializados?

2) Por que não beneficiar um grupo de usuários com todos os ajustes identificados de uma vez?

A primeira questão depende de algum atributo para selecionar o ajuste a aplicar primeiro; além disso, poderia ter impacto negativo na análise dos outros ajustes pois, dependendo do *website* avaliado, é possível que a porcentagem de visitantes que retornam seja baixa e, consequentemente, outros ajustes contem com cada vez menos participantes (observações). Dessa forma, estes aspectos nos levaram a optar pela aplicação de todos os ajustes identificados, considerando todas possíveis utilizações de ajustes para que assim a ferramenta possa contar com todos os dados possíveis para avaliação dos ajustes.

Outra dúvida surgida durante o trabalho de tese envolveu definir quando e onde aplicar os ajustes. Por exemplo:

1) Apenas uma observação é suficiente para servir de base para a aplicação de um ajuste?

2) Dado que os dados capturados estão relacionados com uma determinada URL, deve-se aplicar um ajuste apenas para essa URL ou deve-se tentar aplicá-lo em todas URLs do *website* avaliado?

A forma de lidar com a primeira questão também reside no fato de que visitantes que retornam podem não ser tão frequentes quanto se deseja para identificar um padrão de uso ou até mesmo uma curva de aprendizado no uso de uma determinada página Web. Sendo assim, em *websites* com uma baixa porcentagem de visitantes que retornam, poucas observações seriam obtidas e, consequentemente, a porcentagem dos ajustes bem sucedidos tenderia a ser menor. Em relação à segunda questão temos que elementos usados em uma determinada página e endereçados por um determinado ajuste podem se repetir ao longo de todo um *website* (e.g., navegação, barra de ferramentas de acessibilidade, rodapé das páginas Web). Dessa forma, a solução adotada considera que todos os ajustes gerados para um determinado *website* serão passados ao módulo cliente que, por sua vez, é responsável por verificar se é possível aplicá-los à URL visitada pelo participante que retornou.

Outra questão relacionada aos ajustes diz respeito a como aplicar ajustes em elementos de IU que possuem papeis semelhantes (e.g., botões cancelar e enviar de um formulário). Na abordagem proposta é possível que um dos botões seja ajustado e tenha seu tamanho incrementado (e.g., o botão enviar). Dessa forma, a solução de design poderia ser vista como inconsistente em relação ao tamanho desses elementos de IU. Entretanto, a abordagem é fortemente baseada no uso; se um determinado elemento requer ajuste para facilitar sua ativação, a ferramenta o fará. Ainda, a avaliação contínua pode resolver eventuais conflitos ao longo do tempo (e.g., aumentando primeiro o botão mais utilizado – enviar – e, em um segundo momento, o outro botão – cancelar). Por fim, outra forma de lidar com esta questão é que o WELFIT 2.0 possibilita que os avaliadores verifiquem qual será o resultado dos ajustes durante o cadastro de normas, dando ao avaliador poder de decisão de quais serão os ajustes aplicados pela ferramenta.

A definição dos ajustes utilizados pelo WELFIT 2.0 foi baseada em padrões de uso identificados em estudos desenvolvidos ao longo do projeto da tese. Esses padrões comumente tinham relações com ações repetidas. Entretanto, a abordagem apresentada nesta tese não é dependente dessas repetições. Qualquer padrão a ser identificado que possa ser convertido em uma sequência de eventos pode ser considerado nos ajustes e, consequentemente, na abordagem autoajustável apresentada.

Por fim, a abordagem seguida considerando um ajuste referenciando elementos abstratos possibilita que uma mesma norma seja aplicada via diversos ajustes concretos para a mesma página de um *website* ou até mesmo para *websites* diferentes. Neste ponto esta proposta se diferencia de trabalhos que utilizam modelos de tarefa, pois cada tarefa tende a ser específica e dificilmente possibilita aplicação em páginas Web ou *websites* diferentes. Complementarmente, a manutenção das tarefas nessas ferramentas é custosa. Já na abordagem apresentada nesta tese, se o *website* é alterado, um novo ajuste para os novos elementos de IU será gerado e experimentado, reforçando o caráter de avaliação contínua defendido nesta tese.

7.2 Limitações do estudo

Em relação às dificuldades encontradas durante o desenvolvimento do projeto, os seguintes pontos foram levantados:

1) O tipo de análise proposto conta com resultados mais expressivos e consequentemente ajustes mais bem sucedidos quando o estudo envolve um *website* com grande visitação ou com alta taxa de visitantes que retornam, uma vez que somente uma parte dos visitantes que retornam está na parte experimental ou controle do *cluster* e as observações realizadas após o ajuste são fundamentais para a análise dos ajustes;

2) O *website* estudado não tem pretensão de representar toda a diversidade da Web. Entretanto, apoiou a validação da abordagem apresentada de autoajuste de IU através da manipulação de tamanho de elementos de IU, alteração de contraste e reposicionamento de elementos de IU para pessoas de usam TA;

3) Neste estudo foram considerados ajustes relacionados à estrutura e ao *layout* de páginas Web. Nenhum ajuste relacionado ao conteúdo foi considerado, dado que processamento de linguagem natural seria necessário para manipular conteúdo e, dessa forma, este tipo de ajuste está fora do escopo considerado nesta tese.

7.3 Trabalhos futuros

Entre os trabalhos futuros temos a aplicação do WELFIT 2.0 em outros *websites* e outros contextos de uso. Outro ponto de interesse é análise subjetiva dos ajustes aplicados, considerando como os usuários percebem os ajustes, se eles os notam, se atribuem melhora ou piora, entre outros valores.

Considerando o uso do d_{SAM}, trabalhos futuros envolvem a utilização de outras soluções envolvendo pesos diferentes para as operações envolvidas no cálculo do d_{SAM}, a atribuição de pesos diferentes para eventos mais significativos em relação à interação (e.g., *click*, *submit*) e a mensuração da distância entre diferentes cadeias de eventos considerando Múltimo Alinhamento de Sequências.

Em relação ao mecanismo usado para converter seletores das normas em máquinas de estado, uma possível melhora é tornar o seletor e a máquina de estados mais poderosos para que a identificação das sequências desejadas nas cadeias de eventos possa ser mais detalhada. Outro ponto que requer investigação é a inclusão de novas normas, por exemplo, envolvendo uso de teclas de atalho e navegação via teclado.

Um ponto de interesse maior para trabalhos futuros é o estudo sobre como determinados eventos de IU, incluindo eventos relacionados a IUs modernas considerando telas de toque e acelerômetro, possibilitam que avaliadores possam inferir interesse por parte dos usuários em determinadas áreas de uma página Web, a partir de determinados padrões de sequências de eventos.

Por fim, um caminho que esta tese abre para investigação é sobre o uso da abordagem autoajustável de IU para centralizar alterações, propostas de *redesign* e experimentos para reduzir o número de incidentes de uso, dessa forma, possibilitando que uma IU se torne cada vez mais adequada aos seus usuários.

Appendix A

A Framework for Web 2.0 Secure Widgets

Vagner Figueredo de Santana, Maria Cecília Calani Baranauskas
Institute of Computing – University of Campinas (UNICAMP)
Albert Einstein Av.,1251, Campinas, São Paulo, Brazil

Marco Aurélio Amaral Henriques
School of Electrical and Computer Engineering – University of Campinas (UNICAMP)
Albert Einstein Av.,400, Campinas, São Paulo, Brazil

Abstract: The evolution from Web 1.0 to Web 2.0 brought possibilities of sharing information with a power and reach never seen before. However, this power of sharing content also brought a number of possibilities of spreading malicious code. This work presents common threats regarding JavaScript usage in the Web 2.0 and proposes how to develop secure widgets considering contemporaneous technologies. From the presented proposal one expects that Web developers could reuse the framework in order to avoid attack vectors combined with the power of sharing and spreading information present in the Web 2.0 and not requiring end users to perform any action.

Keywords: Secure widget, Cross Site Scripting, XSS, Cross Site Request Forgery, CSRF, Secure JavaScript

A.1 Introduction

The Web and its technologies are constantly in evolution and this was shown in the transition from Web 1.0 to Web 2.0. In the Web 1.0 the paradigm of one content producer to many consumers that can access content at any time was a revolution when compared to the synchronous mass media. This change occurred due to a number of factors, for instance, emerging technologies and the usage that people were making of the Web.

At late 90's the Web became a communication technology where everyone can contribute with, consume, produce, and publish content; using any type of media, at any time. This significant change is one of the forces that lead to the creation and wide use of the term Web 2.0, which is an interactive environment that has as main characteristics the integration of technologies of information and communication, supporting cooperation and content sharing.

In the Web 2.0 the development of code interpreted at the server deals with the filtering of requests between browsers and servers using authentication and other resources to avoid the execution of malicious code. In this case the variables related to the security and compatibility are significant, but the developers' concerns are related to the server configuration and requests format.

When considering client-side development for the Web 2.0, the heterogeneity of devices and browsers makes the task of developing compatible and secure client-side code (i.e., avoiding the execution of malicious code) a challenge. To illustrate this fact, it is worth mentioning that in the top 10 security vulnerabilities of Web applications, two of them are strongly related to the JavaScript language usage at client-side, namely: XSS (Cross Site Scripting), the 2nd, and CSRF (Cross Site Request Forgery), the 5th [82]. These vulnerabilities are related to vectors generated by problems in the client-side code that are apparently neglected due to the incidence in the Web 2.0. In addition, these vulnerabilities are related to the technologies used in the Web 2.0 and to how hard is the verification of JavaScript code when there is cooperation and sharing of data across domains.

JavaScript is present in 96.9% of most popular homepages [131]. Since JavaScript is the script language used in Web browsers and Web browsers became the dominant application to consume content, JavaScript became the most popular programming language. However, its potential was neglected for a long time for many reasons, but AJAX (Asynchronous JavaScript And XML) gave JavaScript a second change [30].

AJAX can be seen as a key component of Web 2.0 applications [81]. Consequently, JavaScript is playing an important role in Web 2.0 applications; this fact brought visibility to attack vectors related to JavaScript coding problems.

Widgets are components offered by many websites that share features and multimedia content. Some of these widgets use JavaScript to send/receive content or even to run scripts at the client-side. However, attackers (i.e., any person trying to use more privileges than s/he is allowed to) can use the power of sharing data of the Web 2.0 to spread malicious code if any attack vector is available.

According to Oda et al. [79], while the massive exchange of static data can be acceptable, restrictions must be imposed when exchanging executable code. Yue and Wang [131] argue that inserting external scripts inside the main document of a Web page is dangerous because it stays at the same scope that other scripts and allows total control over the page and browser's window. This kind of inclusion of scripts allows access to restrict information (e.g., cookies), access to JavaScript variables, to redirect users, to deface Web pages, or to include data-logger, among others.

Mashup is a new way of building Web applications. It is a way of combining content, programs, and services of many sources in a graceful way, resulting in a new and integrated experience for users. Mashups use the browser as if it was a Lego® platform, allowing the creation of interesting things from existing things [60, 29]. In other words, mashup can be seen as a website that integrates widgets from different sources.

Widgets are been used and shared among an increasing number of websites. Widgets make possible for websites to share content and to stimulate collaboration, among users. However, widgets also bring issues related to security and privacy; point when commonly developers are lead to choose between security and feature [60].

Considering the popularity of the JavaScript language, 74.9% of the most popular homepages of the world use JavaScript in an insecure way. These insecure practices are related to the insecure generation and insertion of JavaScript. Example of insecure insertion is to use the src attribute of the $< script >$ to insert an external JavaScript file into the main body of a Web page. Example of insecure generation is to use data received from users or other sources/domains to generate code via $document.write()$ or $eval()$ functions. More than 43% of the most popular homepages use JavaScript code coming from three or more external domains, only 3.6% of the Web servers deliver JavaScript files via HTTPS (HyperText Transfer Protocol Secure) and only 5.8% of the popular homepages use JavaScript files delivered using HTTPS [131]. Moreover, in average, each website includes content coming from five or more external domains [79]. These data show how preoccupant is the current scenario of Web 2.0 applications in relation to security at the client-side.

In this context, this work aims at presenting common threats related to the Web development regarding JavaScript, a literature review of proposed solutions (and its limitations as well), and a framework to develop a security widget considering the presented scenario and the shortcomings found in literature.

The paper is organized according to the following structure: the next section presents issues related to the communication using JavaScript, section A.3 details common attacks at the application level, section A.4 reviews solutions present in the literature that combine the use of JavaScript and security, section A.5 details the proposed framework for Web 2.0 secure widgets, and section A.6 concludes.

A.2 Data Exchange Using JavaScript

AJAX data transfer was projected so that the data exchange occurs only between Web pages and the Web server that served them (i.e., same-domain). However, in many cases it is desirable to count on mashups sending/receiving data to/from different domains (i.e., cross-domain).

The restriction that avoids cross-domain connections is called Same Origin Policy (SOP). It prevents documents or scripts loaded in one page to obtain or to change properties of a document coming from another domain [97]. The motivation for SOP is that when including script of another domain, there is no way of verifying whether the script is using appropriately the access to the main document [60]. Part of the challenge of dealing with malicious JavaScript code is that they can be dynamically generated, which makes difficult the analysis of static code, since it is hard to determine precisely what scripts are generated and run [131]. Phung et al. [87] present the following code to illustrate this characteristic:

```
var url = { toString: function(){
    this.toString() = function(){ return ‘‘bad’’ ; } ;
    return ‘‘good’’ ;
  }
};
```

In this example the code returns different values for the same *toString*() method because the method overwrites itself after the first execution. In addition, since mashups count on data from diverse sources, they can perform some workaround to avoid SOP and to obtain data from external sources. Moreover, SOP does not apply to $< script >$ tag, where scripts can be downloaded from other domains and run with total access in the Web page that includes them; other examples of cross-domain tags are: $< style >$,$< iframe >$, and $< img >$ [60].

One way of dealing with the restriction presented would be to use a proxy at the server, since server-side programming languages allow cross-domain connections. This solution reaches its goal, but counts on two negative points:

- Bearing in mind the Web 2.0 context in which a widget is used in different websites, all the websites administrators reusing these widgets have to configure the server proxy;

- The website's Web server reusing these widgets will have a significant overload because all the communication has to pass first to the server of the website reusing the widget [100].

Another significant point is the asynchronicity, since a mashup built at the server requires all widgets to be completely loaded; the mashup built at the client allows the content of each widget to be loaded independently, reducing the response time and, consequently, increasing the usability.

One easy way of implementing a mashup at the client with cross-domain capability is by using a technique called IFrame Proxy [34] or Fragment Identifier Messaging [60]. This technique is based on the fact that a Web page that has an $< iframe >$ element has access to the URL object of the $< iframe >$; this means that the main document can insert information, for instance, after the "#" symbol. Therefore the document loaded inside the $< iframe >$ can retrieve information in its URL and guide its functioning based on this information. This solution is interesting and simple, but counts on a usability problem, since some browsers (e.g., Internet Explorer) play a sound at each refresh [100].

Signed Script is an initiative of Mozilla project and involves the generation of a digital signature associated to a certain script in order to give privileges as the access to file system or cross-domain requests using XMLHttpRequest object [100]. Noureddine and Damodaran [78] suggest this solution as a good practice, but it is not consolidated if we consider different browsers and, in such heterogeneous context that the Web is used, this solution can not be considered a practical one.

JSONRequest is a proposal based on the data exchange using JSON (JavaScript Object Notation), a subset of JavaScript used to exchange structured data. Thus, JSONRequest is a solution proposed by Crockford [28] that allows the data exchange across different domains without exposing users to issues related to the execution of JavaScript, since they transfer only JSON data, without sending cookies or any other credentials in the HTTP requests. Beyond JSONRequest Crockford [29] also proposes the < *module* > tag that would be a way of dividing a Web page in a collection of modules. These modules and the website containing them use JSON to exchange data.

One solution that tries to deal with the restrictions, following policies, and being a practical alternative is the Dynamic Script Tag. It is based on the principle that the code contained inside a < *script* > tag is run as soon as the browser loads it. This technique involves the dynamic creation of < *script* > tags, which allows cross-domain data exchange. However, if the content is executable, it can be characterized as an insecure way of using JavaScript, bringing also all attack vectors related to this practice; this will be detailed in the next section.

A.3 Common JavaScript Attacks

XSS (Cross Site Scripting) can be defined as an attack in which a malicious user (or attacker) runs a script inside the website of others. As soon as the script is run, it will have access to all page's variables, cookies, and other information, allowing the attacker to stole sessions, redirect users to other websites, scan the local network, deface Web pages, install data-loggers to obtain confidential information and send them to a malicious server, among others. In addition, XSS Hole can be defined as an entry point or an attack vector for XSS.

A sensitive point related to security of Web pages refers to the insertion of content. It is common for developers at the server-side to focus on how to filter content insertion to avoid code injection (e.g., SQL Injection). At the client-side this concern must also be considered when users insert content that is not stored in data bases (e.g., a term just searched and presented at a results page) or when the content is recorded in a data base (e.g., comments stored in a Content Management System). One way of verifying if the data is correctly filtered in forms is to add codes and HTML (HyperText Markup Language) tags to observe how the Web page shows these codes, i.e., if the tags are filtered, converted, interpreted, etc.

The lack of filtering user data potentially creates attack vectors for XSS that can be Reflected XSS or Stored XSS. Reflected XSS uses content inserted by users in a Web page, for example, resultant page of a search. Thus, if one inserts the content $< script >$ $alert(“XSS'') < /script >$ in a Web form and the page returns the interpreted code, then it is a sign that you have found a XSS Hole. In the Stored XSS the malicious code is recorded at the data base of the attacked website in such a way that when the malicious code is presented to users then scripts perform some cross-site attack. For instance, a forum that does not apply the appropriate filter in messages [121]; See [95] for more examples.

According to Wolf-Iszaevich [129], avoiding XSS is of key importance and this task is harder to perform than avoiding SQL Injection, even that, at first sight, it can look less critical considering security.

When using widgets, an example of insecure JavaScript that opens possibilities for XSS attackers is the usage of the function *eval*(), function that runs the code passed as parameter. Thus, if the parameter is not correctly filtered, then it is possible to insert malicious code in a similar way occurred with Reflected XSS. In addition, 44.4% of popular homepages use *eval*() [131]. Example of insecure usage of the *eval*() function:

```
var response = eval(o.responseText) ;
```

In this example the content received as response is being run and the result stored in the response var. Hence, if there is any malicious code in the response, then this script is at the same scope that the *eval*() function and, as a result, can access all variables, cookies and other information.

One example of privacy attack using information leak is the acquisition of cookie content using attributes of cross-domain tags to send data to other domain. Suppose that an attacker has access to the main page script (e.g., via XSS Hole), then s/he can use the following code:

```
var image = new Image() ;
image.src = ‘‘http://www.example.com/logger?c=encode(document.cookie);
```

Consequently, any information that can be read in the document can be sent in a similar way, including confidential information logged as password, credit card number, etc. [79].

CSRF (Cross-Site Request Forgery) is a technique used by attackers that considers vulnerabilities of the browser and the target website [78]. It uses the fact that in certain cases browsers insert credentials as cookie and IP address in the requests. Thus, if an application authenticates users using only these credentials (that can be accessed at the client-side), without any verification at the server, then it allows this kind of attack. Hence, if an attacker finds a way of accessing the rendered code (e.g., via XSS Hole), then s/he can use the mentioned credentials to send requests as if s/he was the authenticated user.

CSRF also occurs when a user visits a Web page that is based on URL to execute certain actions and allows attackers to perform request using user privileges to access this kind of URLs, without the user's consent [79]. Example of CSRF via src attribute of the $< img >$ tag:

```
var image = new Image();
image.src = ‘‘http://www.example.com/reportabuse/1234’’;
```

In this example the creation of the image object results in a request using the credentials of the authenticated user. Thus, in an online social network, for instance, the code inserted in different accounts would result in a coordinated attack against a certain user, suggesting that owners of these accounts reported abuse of system usage.

A.4 Secure JavaScript Literature Review

Literature has proposals of technologies, libraries, and tools to mitigate common threats of insecure use of JavaScript, summarized as follows:

Subspace is a communication primitive among components of a Web page and depends on the manipulation and control of the property document.domain with the nesting of $< iframe >$ tags in such a way that the main Web page communicates with a mediator

$< iframe >$ that, in turn, communicates with an $< iframe >$ it contains, which uses external content [60].

SOMA (Same Origin Mutual Approval) is a policy to control information flow avoiding common vulnerabilities as XSS and CSRF. The idea is to make SOP more restrictive so that a mashup works only if all websites involved explicitly allow their participation. SOMA can prevent diverse vulnerabilities of Web applications. However, it requires an extra HTTP connection for each request and the definition of configuration files to specify policies of content exchange among domains in the widget provider and in the domains that use the widget [79]. The solution is interesting, but not feasible in practice because it would be implemented in browsers or used as add-ons. Moreover, to obtain all of its benefits, developers using/maintaining mashups would have to define the list of trusted domains. SOMA does not prevent attacks coming from a XSS Hole. It avoids data leak to untrusted sources.

NoScript is an add-on for the Firefox Web browser that has a content blocking mechanism considering interpreted or compiled code (e.g., JavaScript or Java, respectively) [65]. It allows the user to indicate from which domains the browser is allowed to run code, in a similar manner as popup blockers. The proposal is interesting; however, the responsibility of configuration is in the users' hands. In addition, bearing in mind that popular homepages reuse content of five or more external domains in average [79], this solution can have negative impacts on users' satisfaction. Finally, it is a solution for one Web browser and not a practical solution for all Web 2.0 applications.

Lightweight Self-protecting JavaScript is an approach that aims at controlling and modifying the behavior of JavaScript code to make it protect itself. It is based on the concept of reference monitor, a method to specify and implement information systems using a component that intercepts relevant requests related to security and then apply policies to decide which requests have to be answered. Regarding implementation, it uses Aspect Oriented Programming to define interception points related to function calls being monitored [87]. The outcome of the technique is that even with problems in the coding allowing XSS attacks, the security policies still can be applied. However, considering the current JavaScript engines implementations, this technique is vulnerable to an attack using the delete function, in which the wrapper functions can be deleted and then the core functions are again exposed, without the mentioned monitoring.

Santana et al. [106] present a survey of guidelines for Web developers regarding technologies and common threats related to the use of insecure JavaScript. In addition, they present some test routines to consider code maintenance. However, they do not present how to combine these guidelines into a practical solution or rationale to develop a secure widget. The guidelines are summarized as follows:

1. Avoid the insertion of external scripts;

2. Use HTTPS to exchange content between domains;
3. Exchange data instead of scripts;
4. Wrap up external scripts in components of limited scope;
5. Filter any code/content inserted by users.

They also present a case study considering how evaluation tools require certain behaviors allowing the functioning of JavaScript data-loggers and how these tools are interesting targets for attacks, since they commonly have JavaScripts running in (potentially) huge number of websites. As an example, the Google Analytics had in 2005 more than 230,000 registered accounts [93].

A.5 A Secure JavaScript Widget Framework

Applications are significantly ahead of Web browsers' technology [29]. Solution proposals for the presented attacks are far from being a consensus, since they depend on the implementation in all common Web browsers [60]. Noureddine and Damodaran [78] suggest as a good practice to disable JavaScript to avoid attacks. However, disabling it means blocking the use of Web 2.0 websites the way we have nowadays.

The rational of the framework considers that:

- The functioning of JavaScript makes hard the task of filtering JavaScript using only JavaScript, as presented in [87];
- External content inserted into a Web page must have controlled permissions, in other words, wrapped, filtered, etc., as discussed in [131];
- Lastly, the page or communication represent points of attacks, thus, XSS Hole and any injection in the communication must be avoided, as presented in [106].

Hence, we propose a framework to develop secure JavaScript widgets taking into account the Web 2.0 context, avoiding the presented attack vectors. The framework (Figure A.1) follows patterns from PoEAA (Patterns of Enterprise Application Architecture) [40] and GoF (Gang of Four) [41] catalogs. The framework is architecturally organized following the MVC (Model-View-Controller) architecture, having View and Controller components. Moreover, the framework involves the combination of a client-side filter, following the Proxy GoF pattern [41], and a widget rendering library, following Template View PoEAA pattern [40]. The presented components may be refactored, but for the sake of brevity we present an overview of the solution and its rationale. Finally, the communication and cross-domain data exchange is performed using only content data format (e.g., JSON, XML).

Requirements of the framework are:

- The Web page must be free of any XSS Hole because the presence of any allows attackers to change the callback and any rendered component of the framework;
- The website must be free of insecure JavaScript generation or insertion, since any of them can result in attack vectors as presented in the section 3.

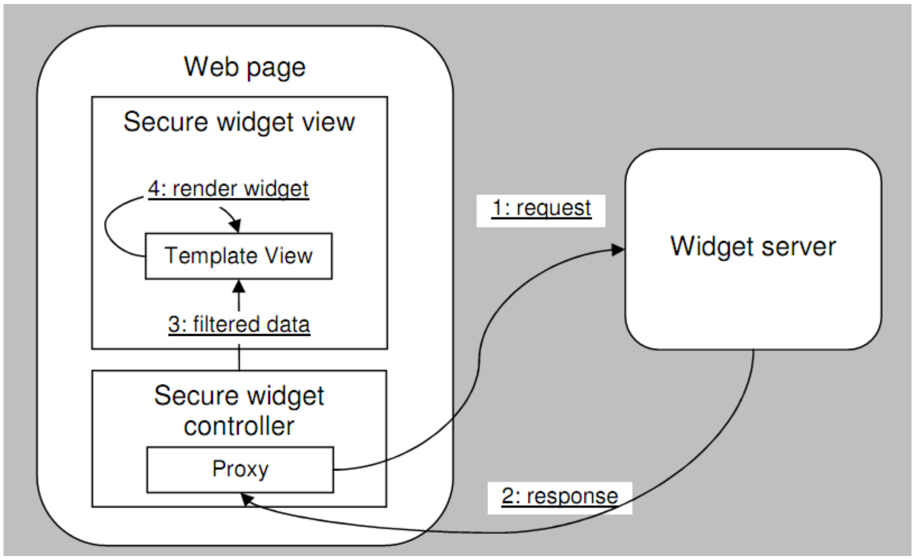


Figure A.1: Overall architecture and message exchange of the proposed framework.

The responsibility of the Proxy component is to mediate cross-domain requests and guarantee that the content passed to the script at the main Web page scope is not code to be run. The Proxy can be implemented in technologies allowing cross-domain data exchange (e.g., Flash or Applet Java) and communication with JavaScript. A key characteristic of this component is that it must not run JavaScript code received. The component then assures that JavaScript malicious code is not inserted in the Web page using the widget. Then, when the Proxy receives the data, it applies verifications and then passes the content to the Template View.

The Template View component is used to embed proper user interface component built considering the data received.

The framework must consider the integrity of the request (Figure A.1, message 1). The data retrieved can be a point of a privacy attack. Thus, in order to complete the solution, we propose the usage of a HTTPS connection to assure message integrity and authenticity, so that the framework avoids the execution of malicious code.

The validation of an instance of the framework is illustrated in Figure A.2, considering current technologies and the intents and solutions of patterns followed. The messages 1 and 2 (Figure A.2) are under a HTTPS protocol assuring integrity and authenticity. Then, the filter's responsibility is to eliminate any JavaScript code that can be run, resulting only in data for the renderer. Therefore, the widget is rendered assuring that no JavaScript code is injected nor the index.html receives malicious JavaScript code. The final remark is that if the page contains any XSS Hole, then the framework will not work, as it would occur to any application using secure transfer (e.g., via Secure Socket Layer).

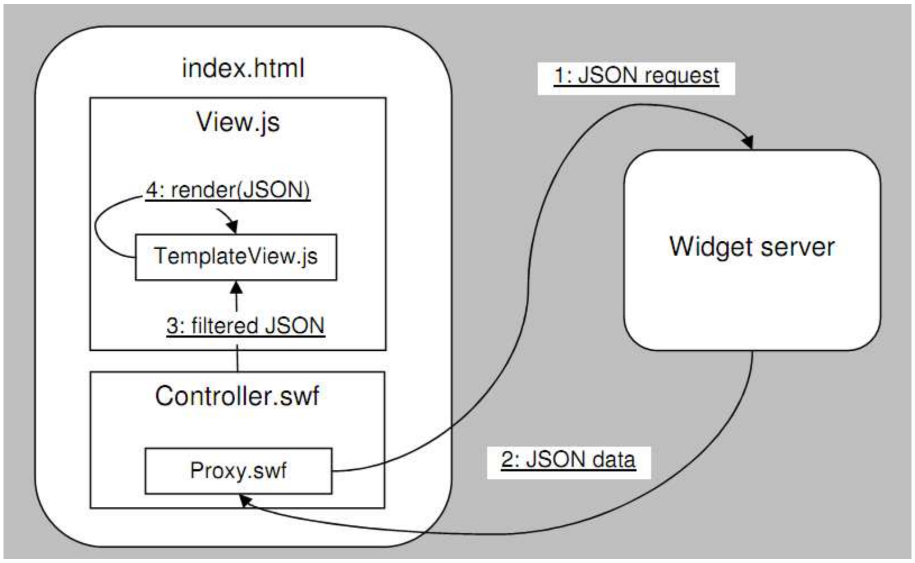


Figure A.2: Instance of the proposed framework considering current technologies.

A.6 Conclusion

The integration of media is changing significantly the way people interact with information systems. Moreover, data show that developers are finding workarounds for security policies in order to use certain cross-domain functionalities present at the Web 2.0.

The popularity of JavaScript, its usage in cross-domain data exchange, inconsonant implementations of solutions, the incidence of XSS and CSRF attacks, and the limitations pointed out in the literature review reinforce that it is hard to automatically verify JavaScript code used in widgets and mashups. This occurs either by the proper nature of the language hardening the task of verifying code or by the security model of Web browsers.

In this context, this work proposed a framework for developing a secure JavaScript widget in such a way that the widget servers must only assure the availability of the request/response using a technology guaranteeing integrity and authenticity (e.g., HTTPS). Thus, Web developers can build the solution considering existent technologies, assuring the security of websites without requiring any action from users.

Finally, client-side programming is not less or more important than server-side programming; we tried to present in this paper that it is just another part of the Web 2.0 applications and should also be considered in designing information systems requiring security and privacy.

A.7 Acknowledgement

We thank FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) for supporting this research (grant #2009/10186-9).

Appendix B

Web Accessibility and People with Dyslexia: A Survey on Techniques and Guidelines

Vagner Figueredo de Santana
IC – University of Campinas
Albert Einstein, Av. 1251, Campinas, SP, Brazil
vsantana@ic.unicamp.br

Rosimeire de Oliveira
CMCC – Federal University of ABC
Santa Adélia, St., 166, Santo André, SP, Brazil
rosimeire.oliveira@ufabc.edu.br

Leonelo Dell Anhol Almeida
IC – University of Campinas
Albert Einstein, Av. 1251 Campinas, SP, Brazil
leonelo.almeida@ic.unicamp.br

Maria Cecília Calani Baranauskas
IC – University of Campinas
Albert Einstein, Av. 1251 Campinas, SP, Brazil
cecilia@ic.unicamp.br

Abstract: Although the dyslexia has significant occurrence in the global population, ranging from 15 to 20%, not much is known about how developers, designers, and content producers should respect differences and consider people with dyslexia in the Web. In this paper we present a survey regarding the state of the art on dyslexia and Web Accessibility. From the results, we present a set of 41 guidelines that may support website stakeholders (i.e., people directly involved with the design, development, and content) in phases involving design, coding, and Web content insertion. Moreover, we propose a mapping of these guidelines considering the responsibilities of different roles of websites stakeholders. Informed by this survey we expect development teams to objectively consider abilities of people with dyslexia in order to remove accessibility barriers.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces – Evaluation/methodology. D.2.2 [Software Engineering]: Design Tools and Techniques – User interfaces.

General Terms: Design, Human Factors.

Keywords: Web Accessibility, Dyslexia, Guidelines, Reading Disability, Learning Disability, Legibility

B.1 Introduction

Content sharing in the Web is growing significantly in the last decades (just for a glimpse, Facebook reported 5 billion shared items on January 2010[1]). This fact has impacted on cultural changes in the worldwide population. When people start sharing or consuming content through information systems, accessibility is a key term to be taken into account.

Accessibility can be considered in a number of contexts, for example, architecture, communication, services, among others. Web Accessibility means that all people, regardless the different kinds of limitations, can perceive, understand, navigate, interact, and contribute to the Web. An accessibility barrier is defined as anything that makes harder or impossible for people with disability to use the Web [2].

For the organizations, accessibility is a key factor to be considered in their websites, since it is directly related to profit, legislation, and the image that the organization has within society. Considering the users' point of view, Abascal and Nicolle [4] say that computers potentially contribute increasing the social inclusion and autonomy of people with disability, since they provide access to education, work, information, communication, leisure, etc.

[1] http://www.insidefacebook.com/2010/02/15/new-facebook-statistics-show-big-increase-in-content-sharing-local-business-pages/

Being a citizen is becoming more and more dependent on the fact of being digitally literate and of having access to the Web. In addition, different modern cultures depend on ICT (Information and Communication Technologies). Example of this dependence is that several government, educational, and entertainment services are offered only online. Moreover, working positions that do not require the use of ICT are becoming rare. Unfortunately, efforts to guarantee accessibility of websites are not yet growing accordingly.

When thinking of Web Accessibility, issues that arise first are commonly related to visual disabilities, since common barriers are related to how to present visual information in a non-visual way. Questions related to other types of disability are hardly considered (e.g., reading disability). Dyslexia is not often included in studies involving Web Accessibility.

According to the Diagnostic and Statistical Manual of Mental Disorders-IV (DSM-IV) [6], the dyslexia (or reading disorder) is characterized by an inferior ability in reading in comparison to the expected for the chronological age.

Regarding the incidence of dyslexia in the population, the International Dyslexia Association (IDA) [55] asserts that from 15 to 20% of the world population have some symptom of dyslexia as slow reading, imprecise reading, difficulty when spelling, low proficiency in writing, or tendency of exchanging similar letters.

Beyond presenting percentages as motivators to highlight the need for actions regarding dyslexia in the Web, this work has as main basis the Universal Design principles. This means that we aim at dealing with dyslexia in such a way that solutions do not segregate this group of users.

McCarthy and Swierenga [67] highlight that much work has been done in order to relate dyslexia to Web Accessibility, but there is no significant number of works yet aiming at the articulation of both, Web Accessibility and dyslexia. Most efforts are focused on how to support users that are blind or have low vision.

The objective of this work is to present a survey regarding the state of the art on accessibility and people with dyslexia, aiming at removing or avoiding barriers that they may face in the Web. Based on this survey involving literature results, techniques, and guidelines, we compiled guidelines according to actions and responsibilities of developers, designers, or content producers. This role-based organization of the guidelines was inspired by the work presented in [101] and aims at providing filters so that website stakeholders can easily apply guidelines that are closely related to their roles.

This work is organized as follows: Section B.2 summarizes the concept of dyslexia and questions involved in dyslexia diagnosis. Section B.3 shows the background to the work. In the sequence, Section B.4 presents related works and details the literature review. Following, Section B.5 summarizes and discusses the guidelines and techniques compiled. Lastly, Section B.6 discusses the survey and Section B.7 concludes and indicates further work.

B.2 Understanding Dyslexia

The term dyslexia has a Greek origin in which "dis" means disorder and "lexia" language. Thus, dyslexia is a language disorder [47]. The concept of the dyslexia is controversial even among the specialists in the field [98]. The currently more accepted neurological model has the dyslexia as a neurological disorder of genetic origin with persistence throughout life [90].

The World Federation of Neurologists [127] defines dyslexia as a disorder in which a child does not develop the skills of reading, writing, and spelling expected according to their age and intellectual performance, despite having access to regular schooling.

According to the neuropsychological approach, the difficulties are understood as a consequence of one or more dysfunctional brain systems involved in learning [99]. Sternberg and Grigorenko [118] proposed a definition that became one of the most used. It is a multiple causes approach that suggests that the reading difficulties are the result of an interaction between biological, genetic and neuropsychological, cognitive, and social skills as well. Thus, authors understand dyslexia as etiologically heterogeneous phenomenon.

In relation to etiological factors related to reading difficulties there are two main lines of discussion: a biological (genetic and neuropsychological) and a functional (memory, phonological awareness, connection of written form to the spoken form, learning and recognition of rules/exceptions, and linguistics inferences) [47, 98].

According to the DSM-IV, people with dyslexia have difficulties to understand written words and sentences in such a way that these difficulties interfere significantly in the accomplishment of common tasks that require reading abilities. Dyslexia is more commonly found in men, representing from 60 to 80% of the population that has this disability [6].

The IDA [54] defines dyslexia as a language learning disability of neurological basis that results in linguistic difficulties, especially those related to reading, but it may also interfere in other abilities such as orthography, writing, and pronunciation. The effects of dyslexia vary depending on the level of disability and the intervention offered to the person.

However, this disability should not be directly related to intellectual levels Dyslexia is not related to low intelligence or educational attainment. Dyslexia is not itself a disease, it is a dysfunction of the brain related to language processing. It is a specific impairment in reading accompanied by a normal development in other areas that are not related to the disability. Some authors understand it as a different way of thinking instead of a disease [73, 112].

In order to have a dyslexia diagnosis the person must present the following characteristics [6]: "A) Reading achievement , as measured by individually administered standardized tests of reading accuracy or comprehension, is substantially below that expected given the person's chronological age, measured intelligence, and age-appropriate education. B) The disturbance in Criterion A significantly interferes with academic achievement or activities of daily living that require reading skills. C) If a sensory deficit is present, the reading difficulties are in excess of those usually associated with it."

There are several characteristics employed to refer to people with dyslexia. This way, we are talking about a group that is united by the diagnosis of dyslexia, but, at the same time, is separated by the complexity that the diagnosis itself involves. We can not consider people with dyslexia as a homogeneous group due to lack of consensus regarding a precise diagnosis. Finally, the literature does not count on efficient tools to attest the exact cause of learning disabilities.

When interacting with computers, people with dyslexia can benefit from using assistive technologies as screen readers, voice recorders, and voice synthesizers [110]. Other categories of tools as spell checkers [32] and screen magnification are also valuable.

B.3 Methodological Stance

Universal Design (UD) consists in the design of products and environments that are usable by everyone, in the widest possible extension, without the need for adaptation or specialized design [24]. The difficulty in fulfilling the UD proposal is clearly due to a number of factors including technological and methodological limitations. Nevertheless, UD is considered a goal to be pursued because it motivates the development and design of solutions that respect different necessities without discriminating.

Within the UD domain Abascal and Azevedo [3] define inclusive design as the one which aims at considering the needs of all the users in mainstream applications and not only in the systems especially designed for people with disabilities. This work is aligned with the inclusive design stance.

The inclusive design is presenting signals of overcoming the welfarist view of products design. Now, when one refers to social inclusion also refers to wishes, emotions, autonomy, and social rights of people. Regarding affectivity, it means considering also the individual happiness as a criterion for the definition of citizenship [109]. Thus, this work follows the UD's philosophy and considers the inclusive design in the process of applying this philosophy, since the main goal presented is the design of products and environments that are usable by everyone. Moreover this goal may be achieved by considering the needs of all users in the mainstream application, including people with dyslexia.

Regarding organization of guidelines, in [101] the authors proposed an integrated an objective way of presenting references involving website accessibility and usability. In addition, in that work three roles associate actions to responsibilities of different professionals: developers, designers and content producers, as follows:

- Developers: are people with expertise in Web technologies and programming. Their responsibilities are within modeling/coding systems and algorithms;
- Designers: are people with expertise in designing user interface (UI) solutions involving User Experience and Information Architecture. Their responsibilities are within designing UI, defining color schemes, defining layout, among others;
- Content Producers: are people in charge of writing and publishing content into websites. Their responsibilities are within updating Web pages' content and writing new content for the website.

It is worth mentioning that these roles are not strictly defined and, in small teams, one stakeholder may assume more than one of these roles (e.g., designer and developer, content producer and designer). Some evidences were shown that such structuring helps stakeholders to quickly reach relevant information regarding website maintenance [101].

In this work, outcomes and guidance provided by different associations, academic papers, reference websites, and specialized magazines were summarized and organized to form an objective and straightforward to use set of guidelines.

The method used in the literature review was the following:

1) we first identified and extracted guidelines, results, outcomes, and techniques presented in the academic papers mentioning dyslexia and Web Accessibility;

2) then we grouped all the extracts found according to the papers' sections where they appeared, terms used, and goals;

3) finally we compiled all extracts and rewrote them in a guideline style in order to facilitate their use, placing them under the groups identified. The groups were formed according to the UI element that the guidelines are referring to and are detailed in the guidelines mapping section.

It is noteworthy to mention that the groups found are not a definitive set and they may grow as the study of dyslexia and Web Accessibility evolves. Next we present a summarized view of the related work.

B.4 Related Contributions

The next subsections present works that aimed at studying dyslexia and Web Accessibility. The first subsection presents works of organizations that support people with dyslexia, in the sequence a subsection counts on academic research papers, and then a subsection details existing guidelines.

B.4.1 Organizations that support people with dyslexia

The British Dyslexia Association (BDA) [11] is an institution that has as a main vision a society that is aware of people with dyslexia so that these people can reach their whole potential. The institution promotes initiatives that aim at promoting sound changes that bring benefits to people with dyslexia. Moreover, BDA also offers support to people with dyslexia and supports people that have direct contact with them, looking for innovative solutions and promoting best practices in national and international levels. In this work we refer to the guide offered by the institution that deals with text, accessible formats, and Web design.

Rainger [89] presents a perspective on content production in the Web aiming at the accessibility regarding people with dyslexia. The referred work was published at JISC Techdis , a British website that offers consulting services about technology and social inclusion. The service supports educational institutions and enterprises via knowledge sharing concerning accessibility, stimulating the innovation and offering counseling about technology and disability. The author focus his work on content accessibility and concludes that there is a need for developers to be aware of the diversity of perceptions of content, learning styles, cognitive limitations and learning strategies.

Bradford's concerns are related to how to design Web pages for people with dyslexia [14]. His work is published in the Dyslexia Parents website that counts on the Dyslexia Online Magazine. The website has the objective of supporting parents of children and teenagers by providing information and counseling. The website also provides resources that deal with dyslexia from different perspectives including, parents, students, and teachers. The author concludes his work pointing out that the main goal to be achieved in order to cope with dyslexia is to promote understanding of textual content.

B.4.2 Academic research papers

Al-Wabil et al. [5] present an exploratory study about the experience of people with dyslexia when surfing the Web. The results were obtained from semi-structured interviews conducted with 10 individuals with dyslexia. The research subjects were from 18 and 49 years old. The study investigated needs, challenges, and difficulties faced by the participants. All subjects had been diagnosed with dyslexia and 3 of them also had dyspraxia (one of these 3 participants also had sensorial defensiveness). In the study, researchers also observed with special interest the use of assistive technologies. The referred work contributed by informing about techniques and types of accessibility barriers the subjects face. The authors conclude that they had interviewed a relatively small group of individuals, but the study suggested the presence of patterns in navigation and scanning of Web pages for information by dyslexic users. In addition, the authors point out that despite the existence of websites conforming to accessibility standards, the Web suffers from many problems in supporting navigation and content consumption, particularly for users with specific learning difficulties.

McCarthy and Swierenga [67] contributed with a literature review considering published works dealing with dyslexia and accessibility in order to determine the knowledge produced until 2010. They investigated available information regarding user experiences of people with dyslexia within the Web. They also considered disabilities that can interfere in the overall understanding on how people with dyslexia use the Web. The authors associated dyslexia with disabilities in general, especially the low vision. The paper has objectives similar to ours, but the referred work presents mainly pointers, without showing objectively how to apply guidelines or remove barriers for people with dyslexia. In addition, the work places dyslexia in the background of other disabilities, fact that may hide specificities on how people with dyslexia interact with the Web.

B.4.3 Guidelines

Zarach [132] presents 10 guidelines that were published in the CETIS (Center for Educational Technology and Interoperability Standards) website. The CETIS provides advices referring to technology and education to educational institutions of United Kingdom. The objective of the website is to contribute to the debate about the use of technology in the field of education. Zarach concluded that the presented guidelines aim at enhancing readability and accessibility for people with dyslexia, but they also benefit people without dyslexia, since focus on improving simplicity, clarity, and usability. This is very much in accordance with the UD paradigm.

The WARAU (Websites Adapted to Requirements of Accessibility and Usability) is a reference website where website stakeholders can discuss norms, guidelines, techniques, and good practices for the development of accessible and usable Web code. WARAU's target audience is composed of websites stakeholders that already have some knowledge in coding Web pages (e.g., HTML, CSS, and JavaScript) and that want to learn how to build valid, accessible, and usable websites. In the website there are discussion areas to a number of topics involving accessibility and usability, from assistive technologies to guidelines and standards provided by organizations such as the World Wide Web Consortium (W3C). Although presenting an objective approach, with references and code examples, the website does not present any guideline or technique specifically addressing dyslexia.

Sets of guidelines for the elaboration of accessible Web content contribute significantly to the wide adoption of actions towards the promotion of Web Accessibility. Examples of available sets of guidelines are the Web Content Accessibility Guidelines (WCAG) [123] and the Section 508 [2]. The WCAG is the most adopted set of guidelines and currently it is in the 2.0 version.

When investigating how WCAG deals with dyslexia, one can observe that two external references [12, 89] were used in order to elaborate guidelines. Moreover, dyslexia is directly mentioned in one success criterion (i.e. 3.1.5 Reading Level). The theme of dyslexia is indirectly considered in some guidelines, but, without a direct reference, reading disabilities may be neglected by WCAG readers.

We argue that the lack of explicit consideration of dyslexia specificities in the guidelines make the needs of users with dyslexia unfulfilled, making them more and more unmotivated in using the Web. This fact is significant when considering autonomy, since difficulties in making common tasks also raises preconceptions and stigmatization of people with dyslexia, bringing negative impacts to their self-esteem, making symptoms even worse.

B.5 Guidelines Mapping

This section presents the integration of guidelines, and techniques resulted from the conducted study. The guidelines are grouped according to the topics addressed, aiming at the easy reuse and implementation. Considering the stakeholders' roles presented in [101], Table B.1 presents groups of guidelines and the respective relevance for the roles considered.

The mapping is intended to serve as an index to guide actions considering the mentioned stakeholders' roles. Thus, for example, content producers could easily know where they should focus their actions and which techniques to use to improve Web Accessibility considering people with dyslexia.

The relevance level was attributed to each guideline according to the following rules:

- If the application of guidelines of a certain group (G) depends strictly on responsibilities of a certain role (R), then the relevance level for this group is set as high;

- Else, if the application of guidelines of G depends highly on a role different than R, but also depends on the participation of the R, then the relevance level is set as medium;

- In other cases, i.e., when the application of guidelines of G has minimum or no dependence on R, then the relevance level for R is set as low.

For example, when referring to end user customization, developers are highly involved in applying such guidelines because the development of customization features involves modeling and programming. Moreover, designers are also involved, since they are responsible for defining the proper design of customization features, UI elements, and colors. Finally, content producers have little participation on applying such guidelines, since they have almost no content to be produced. Hence, the guidelines presented at the end user customization group have low level of relevance for content producers.

Guidelines group	**Developers**	**Designers**	**Content producers**
Navigation	High	High	Medium
Colors	Medium	High	Medium
Text presentation	Medium	High	Medium
Writing	Low	Low	High
Layout	Medium	High	Low
Images and charts	Medium	High	Medium
End user customization	High	Medium	Low
Markup	High	Low	Medium
Videos and audios	Medium	High	Medium

Table B.1: Relationship among groups of guidelines and relevance level according to stakeholders' roles.

The guidelines surveyed are organized according to the following groups:

1. Navigation: involves recommendations related to menu structuring, breadcrumbs, index pages, site map, internal search, links, lists, and how to highlight headlines, sections, and key terms;

2. Colors: involve recommendations related to the combination concerning foreground and background colors;

3. Text presentation: recommendations related to text sizes and font types, as well as alignment and animation;

4. Writing: recommendations concerning content writing style that makes easier for people with dyslexia to read content in the Web;

5. Layout: overall recommendations about how to structure the Web page layout;

6. Images and charts: recommendations about whether charts/images are important to support the understanding of the text and in which cases they should be avoided;

7. End user customization: recommendations about UI elements that are customized by users and that have positive results when offered to users with dyslexia;

8. Markup: recommendations related to techniques of markup languages (e.g., HTML and XHTML) that improve accessibility considering people with dyslexia;

9. Videos and audios: recommendations considering the use of multimedia as videos and audios.

B.5.1 Navigation

Navigation menu

N1 – Navigation must be consistent, visible all the time, and must contain simple lists of links avoiding the need of scrolling pages to see all listed items. Difficulties related to the sense of direction are usual for users with dyslexia [5, 14, 67].

N2 – Avoid dynamic menus or menus that use transparency, because users with dyslexia may find hard to contrast them with the background partially visible [5].

N3 – Whenever using images in navigation, since they facilitate the use for people with dyslexia, make sure that images have significant alternative text (via alt attribute) adding a textual description to textually represent images function [132].

Breadcrumb

N4 – Textual breadcrumbs should have text size that allows comfortable reading. Considering redundant information, breadcrumbs can also have snapshots helping in the task of contextualizing and remembering. People with dyslexia prefer using the back button, especially when there are snapshots of visited Web pages [5].

Index page

N5 – Whenever possible, structure index pages considering a logical order involving tasks sequence or structure (e.g., when involving information that can be structured as part-whole), without requiring exclusively the alphabetical ordering. Most of dyslexic people use index page, but they may have difficulties with alphabetic sequences [5].

Site map

N6 – When building a site map, the use of hierarchical trees containing clear texts is suggested. Site maps are useful, but usually they are used when other search possibilities did not succeed [5, 11, 132].

Back and forward buttons

N7 – Avoid using mechanisms that interfere with back and forward buttons functioning. Users with dyslexia use back and forward browser buttons because they are visible and consistent. They are a solution when a click is done by mistake or if users want to freely go forward or backward when need to read again steps performed in a step-by-step task [5, 89].

Internal search

N8 – When offering internal search in your website, optimize it by aiming for the quality of the results, providing self completion, and orthographic verification to point errors and ease correction [5, 67, 89]. Writing aids are useful for users with dyslexia to correctly perform content searching tasks.

Sections header and highlighting

N9 – Avoid using headers all in capitals in order to call users' attention. If the highlight is needed in headers, use text size and bold face for regular capitalized text, since it is easier to read text in regular capitalizing [11, 89].

N10 – Avoid italics because it is harder for people with dyslexia to read italic text if the text size is reduced. Instead of italic text use bold to highlight key concepts. Give preference in using tags that represent semantic information instead of visual, for example, use $< strong >$ instead of $< b >$. The italic text becomes unreadable in comparison to non-italic text. Finally, assure that style sheets do not use italics [11, 14, 89].

N11 – Whenever text highlighting is needed, consider using boxes, border, and background color to call user attention. The reading time is reduced when users with dyslexia easily identify the section related to the text they are reading [32]. In addition, adding colors increase the probability that the information read is retained in a long-term memory [11, 14, 89].

Links

N12 – When writing link labels, structure labels clearly in order to keep them simple. Do not use "click here" [132].

N13 – Content links must indicate which pages were accessed [11]. Presenting clearly what are the visited links improves reading for people with dyslexia and helps on identifying which pages had been visited.

N14 – Use underline only for links and avoid links involving big blocks of text, because it makes harder to read [89].

N15 – Use links in a concise way at the beginning or at the end of phrases in order to ease the reading [11, 89].

Lists

N16 – Use ordered lists (or numbered lists) instead of unordered lists (or bullet list), since ordered lists have additional information of context and people with dyslexia find it easier to follow [132]. Some screen readers speak the number of list items (e.g., list with 10 items), but it is worth mentioning that not all people with dyslexia use screen reader.

N17 – Use white spaces to ease reading. To improve comprehension of list items, use white space to separate marker from text and between items use double spacing [14, 89, 132].

B.5.2 Colors

Background color

C1 – Avoid pure white as the background color, because the white can obfuscate the text for people with dyslexia. A close alternative is the light gray with the following hexadecimal code #FFFFE5. A significant fraction of people with dyslexia is sensitive to the brightness of white background (i.e., scotopic sensitivity) and the text can appear as if it was moving or blurred. Instead of white background use pastel colors in background. For example, dark blue on beige background. Finally, avoid background images and patterns [11, 14, 89].

B.5.3 Text presentation

Text size

T1 – Avoid small text sizes. Small texts slow reading for people with dyslexia. The smaller text size recommended is from 12 to 14px; for printing it should be from 12 to 14pt [5, 11, 67, 89, 132]. Dickinson [32] reported in an evaluation that people preferred font sizes above 12px (the default in that test case).

T2 – Use mono spaced fonts and without serif. Examples are Arial, Comic Sans, Verdana, Tahoma, Century Gothic, Georgia, and Trebuchet. People with dyslexia, but not only them, consider font without serif easier to read, since that serif fonts are more detailed and, consequently, more complex of reading and identifying letters. For example, in the Times New Roman font the letter 'g' appears to be the number 8 [5, 11, 14, 32, 132].

Alignment

T3 – Do not use justified text alignment, since it counts on irregular spacing between words and it is harder to read. It creates visual patterns that are hard to be ignored by people with dyslexia. The different spaces generated distract users and worsen reading [14, 89].

Animation

T4 – Avoid moving text, since movement or blinking complicates reading [11, 89].

Print and download versions

T5 – Offer textual version for pages so that users can download it and print it [11, 132].

Spacing

T6 – Use a spacing line to separate paragraphs and use space between lines of 1.5 to 2 lines of spacing [89].

B.5.4 Writing

Language and writing style

W1 – When writing, be concise. Consider short paragraphs. Avoid complicated language/jargon. Use short, simple, and direct sentences, with small number of chunks of information. Long sentences contain more than one idea and can be divided into shorter phrases. This gives to people with dyslexia the possibility of making a small pause between the sentences [11, 12, 14, 67, 89, 132]. Reading disabilities make hard the recognition of written or printed words and their relations with the correct pronunciation sounds. This process is called text decoding. Decoding must be automatic for people that read fluently. The act of decoding text word by word consumes mental energy needed to understand the text they are reading. Texts that use short and common words and short sentences are easier to decode and usually require a reading ability less advanced than the ability required to read long sentences and uncommon words [123].

W2 – Give instructions clearly. Avoid long explanation phrases. Use active voice instead of passive voice (e.g., use ‘you need to login into the system’ instead of ‘the login is needed in order to use the system’). And avoid double negatives [11, 89].

W3 – When appropriate, use graphics to illustrate complex text or complex ideas [89].

W4 – Do not use hyphenation of words that commonly are presented integrally [89].

W5 – Write considering screen readers because there are people with dyslexia that use this kind of assistive technology. Make good use of punctuation, for example: consider using period at the end of phrases to assist screen readers to perform a small interval; use semicolon, comma, or periods at the end of list items to clearly separate the reading of the items and avoid sequential reading; do not write text all in capitals (without the proper $< abbr >$ tag) because it is possible that screen readers interpret the capitalized text as an acronym and, consequently, spell it out instead of read it sequentially [14, 132].

B.5.5 Layout

L1 – Prioritize the information. Use simple design. Avoid screens with lots of information. A limited quantity of information should be presented, avoiding scrolling and memory overload. In addition, provide sufficient white space among UI elements [67, 89].

L2 – Avoid large columns of texts. The length of lines may have a big impact in relation to readability considering comfort and comprehension. Consider default width for columns from 60 to 70 characters length (as the ones used in printed newspapers) and use fluid design so that the width of the columns can be adjusted to the browsers' window width or different zoom levels [14, 89].

L3 – Strictly avoid horizontal scrolling [89].

B.5.6 Images and charts

I1 – Use images, charts, and pictures to complement textual information. They divide the Web page in smaller chunks and, for people with dyslexia, provide visual stimuli. However, if they are too big or too small, could make reading more difficult [11, 14, 67, 89].

I2 – Use images and icons throughout the text, particularly in links. People with dyslexia tend to consider more images than words. Include also appropriate descriptions in these images so that screen readers can read them [132].

I3 – Avoid moving or blinking images, because they distract people with disability as well as distract any person [14].

B.5.7 End user customization

E1 – The website should be easily customizable by users. Provide features so that users can configure color scheme (background color, text color, and printing colors), font type, and text size. These may improve reading speed. In addition, people have different abilities and different preferences regarding colors, types, and sizes [11, 67, 132]. For examples of how to do such customization, refer to Designing Web Pages for Dyslexic Readers and SeeWord [32].

E2 – Allow users to control presentation using their own style sheets [89].

B.5.8 Markup

M1 – Assure that the website can be read by screen readers, since some people with dyslexia use this type of assistive technology. Thus, guarantee that the markup is valid, provide alternative text, avoid unnecessary tables, and use semantic markup wherever use tables [101, 132].

M2 – Use the < *abbr* > tag to explain abbreviations and acronyms [89].

B.5.9 Videos and audios

V1 – Whenever counting on videos or audios, do not play these media automatically when the page loads. Provide video or audio only under users' requests. In addition, do not rely only on one media to provide content for users [89].

B.6 Discussion

The reviewed guidelines and techniques show convergence, since most of the studied works shares the aim of facilitating the access of dyslexic people to digital information. However, some conflicts still occur. For example, although there is an indication that the fluid design is more accessible, according to BDA [11], stakeholders should avoid starting phrases at the end of the lines. The conflict here is that if the layout is fluid, there is no way of controlling this. Moreover, considering conflict resolution, a promising approach is to consider any fixed configuration pointed by a guideline as a default in conjunction with any other features allowing users to change presentation as they wish.

Among the works reviewed, it was possible to identify that the end user customization plays a central role in accessibility considering dyslexia. Nevertheless, only two guidelines where found regarding this subject. Thus, a deeper study on end user customization is an identified gap that needs to be bridged.

This work focused on guidelines for dyslexia; on the other hand, as we aim at promoting the universal accessibility, we believe that such guidelines should be integrated into general accessibility guidelines as WCAG [123] and Section508 [2]. As WCAG is the most adopted guidelines set and receives diverse contributions from the international community, future work should involve, but should not be restricted to it. At first glance we can identify similarities between WCAG and the reviewed guidelines such as in the case o of the WCAG Success Criterion 2.4.8 Location and the reviewed guidelines N4 Breadcrumbs and N6 Site Map. Also, some of the reviewed guidelines, e.g., N1 Navigation menu (partially covered by the WCAG Success Criterion 1.4.8 Visual Presentation) and N8 Internal Search (partially covered by the WCAG Success Criterion 3.3.5 Help), have the potential to improve the WCAG. Potential conflicts between them also require further investigation.

An emerging challenge for the Web Accessibility is the accessibility of Rich Internet Applications (RIA). RIAs offer dynamics comparable to desktop applications with the benefits of being pervasive. On the other hand RIAs break some well-established concepts from websites navigation as, for example, the behavior of back and forward buttons (mentioned in the reviewed guideline N7 Back and Forward Buttons). Despite the fact of some techniques to overcome those barriers, the diversity of behavior scripts languages can provide is enormous and unpredictable. Thus, since the works related to dyslexia on web accessibility are still restricted to traditional Web pages, future research are necessary to identify which of the already reported guidelines are still valid for RIAs and the possible new requirements – specially those related to the dynamic behavior – for accessible RIAs.

It may be hard to convince part of stakeholders about the importance of some issues, such as the question related to contrast between foreground and background colors. There are generic usability guidelines indicating that we should use black on white, as presented by Nielsen in [75]. However, as previously presented by guideline C1, this can be prejudicial for people with dyslexia. This fact reinforces the need for features that provide color scheme customization, font type, spacing, and text size, for screen presentation and for printing.

Moreover, we verified how important is to know about specificities related to different types of disabilities instead of grouping them with “others”, attitude that may move actions away from the real needs and preferences of people with dyslexia. This vision is encouraged by the UD concept – to respect the differences supporting the widest possible audience.

The individual differences are relevant, especially when we refer to dyslexia.

B.7 Conclusion

The dyslexia is still not well understood and even characterized. As a natural consequence, its specificities have hardly been treated by Web Accessibility standards, as other well known disabilities do. This work reviewed existing guidelines and techniques related to dyslexia and presented a compiled set of guidelines to inform the actions of website maintenance professionals and other interested parties. The guidelines, grouped according to UI elements, synthesize outcomes of previous works in such a way that developers, designers, and content producers could understand the limitations of people with dyslexia and remove or avoid accessibility barriers.

Considering the proposed grouping according to stakeholders' roles, developers are more strongly concerned with markup techniques and end user customization features; the designer's work is more sensible to layout and user interface design, as well as the color scheme definition; content producers should pay special attention to the writing style considered and also to guarantying images' alternative texts.

The focus of this work was on specific needs of people with dyslexia as there is a lack in understanding their functional limitations and the Web Accessibility barriers faced by them. However, our intention is to allow this set of guidelines to integrate other sets of guidelines that promote UD principles.

As future work a user study is planned to explore solutions based on the proposed guidelines with a local organization that works with correlated themes.

B.8 Acknowledgment

We thank *Fundação de Amparo à Pesquisa do Estado de São Paulo* (grants #2009/10186-9 and #2011/06399-7) and Federal University of ABC for supporting this research.

Appendix C

Documentação do WELFIT

C.1 Casos de Uso

C.1.1 Convida participante

Informações características

Objetivo no contexto: Apresentar convite de participação aos participantes do *website* em avaliação.

Escopo: Módulo cliente.

Nível: Tarefa primária.

Pré-condições: *Website* em avaliação utiliza o módulo cliente do WELFIT, está cadastrado no sistema e o JavaScript está habilitado no computador do participante.

Pós-condição de sucesso: Convite é apresentado ao participante.

Pós-condição de falha: Convite não é apresentado e, consequentemente, a captura não pode ser iniciada.

Ator primário: Módulo cliente.

Gatilho: Módulo cliente é carregado no dispositivo do participante.

Cenário principal de sucesso

1. Módulo cliente apresenta o convite de participação com a pergunta "Usa alguma tecnologia assistiva?" acompanhada de radio buttons sim e não, com os botões "Não participar" (à esq.) e "Participar"(à dir.) e com o link "Mais informações" (no final do convite).

2. Participante acessa o convite.

Extensões

2.a - Participante acessa link “Mais informações”: Um breve texto é apresentado ao participante explicando o funcionamento da ferramenta e como a avaliação é feita, incluindo questões de segurança e privacidade.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 5 segundos para o convite ser apresentado.

Frequência: Uma vez para cada participante que visitar o *website* em avaliação, até que ele responda ao convite.

Canal até ator primário: Navegador.

Agendamento

Entrega: Versão 1.0

C.1.2 Responde ao convite

Informações características

Objetivo no contexto: Responder ao convite de participação.

Escopo: Módulo cliente.

Nível: Tarefa primária.

Pré-condições: Convite está acessível ao participante.

Pós-condição de sucesso: Módulo cliente inicia captura de eventos.

Pós-condição de falha: Participante não identifica o convite ou o ignora e inicia a utilização do *website* sem participar da avaliação.

Ator primário: Participante.

Gatilho: Apresentação do convite.

Cenário principal de sucesso

1. Participante aceita o convite.

2. O módulo cliente grava informações (e.g., número de ticket, respostas) do participante em cookie de aplicação e inicia o registro de eventos.

3. A região da interface utilizada para apresentar o convite é então reduzida para apresentar o status da ferramenta (e.g., gravando, parado) e disponibilizar mecanismos para o participante interromper a captura a qualquer momento.

Extensões

1.a - Participante recusa convite: Módulo cliente fica inativo e convite é fechado.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Cerca de 10 segundos.

Frequência: Uma vez para cada par participante-dispositivo que visitar o *website* em avaliação, uma vez que o ticket é mantido em cookie.

Canal até ator primário: Área da página do *website* em avaliação reservada para o convite.

Agendamento

Entrega: Versão 1.0.

C.1.3 Captura eventos

Informações características

Objetivo no contexto: Capturar eventos disparados no lado do cliente.

Escopo: Módulo cliente.

Nível: Tarefa primária.

Pré-condições: Participante aceitou participar da avaliação.

Pós-condição de sucesso: Eventos ocorridos no lado do cliente são capturados e gravados em memória.

Pós-condição de falha: Módulo cliente não captura eventos.

Ator primário: Módulo cliente.

Gatilho: Convite é aceito pelo participante.

Cenário principal de sucesso

1. Módulo cliente captura eventos disparados no lado do cliente.
2. Módulo cliente monta a linha de log correspondente ao evento capturado.
3. Módulo cliente grava linha de log em memória.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Execução em poucos milissegundos.

Frequência: Durante toda a sessão de utilização do *website*, a cada evento ocorrido no lado do cliente.

Canal até ator primário: Páginas Web.

Agendamento

Entrega: Versão 1.0

C.1.4 Compacta log

Informações características

Objetivo no contexto: Compactar eventos capturados e gravá-los em um pacote.

Escopo: Módulo cliente.

Nível: Sub-função.

Pré-condições: Evento foi capturado e gravado em memória.

Pós-condição de sucesso: Linha de log é compactada e gravada no pacote sendo construído.

Pós-condição de falha: Linha de log é gravada sem compactação.

Ator primário: Módulo cliente.

Gatilho: Um evento é gravado em memória.

Cenário principal de sucesso

1. Linha de log é montada com o evento capturado, seja ela a linha i.

2. Grava-se o conteúdo de i em uma variável auxiliar.

3. Se i não é a primeira linha do pacote, então compara-se i com a linha gravada anteriormente no pacote (em memória), seja ela a linha i - 1, e todos os dados repetidos em i são trocados por vazio (i.e., “”).

4. Agora, a linha i pode ser compactada novamente substituindo caracteres que se repetem na linha de log por uma representação desse caractere imediatamente seguido por uma marcação representando o número de vezes que ele se repete. Exemplo: o delimitador de campos pode se repetir várias vezes para campos vazios, então, ele pode ser combinado com um número que representa seu número de ocorrências (e.g., o trecho “,,,,,,,,,,”, de 10 bytes, passa a ser “,{a}”, agora com 5 bytes).

5. i - 1 recebe o conteúdo da variável auxiliar.

6. Calcula-se o tamanho do pacote.

7. Se as i linhas de log formam um pacote de tamanho menor que o limite configurado (default 1kB), a i-ésima linha é adicionada ao pacote atual.

Extensões

1.a - Se linha i é a primeira do pacote: linha de log é gravada na íntegra. Dessa forma, os pacotes são independentes entre si e a ordem da gravação na base de dados é garantida pelo timestamp de cada linha.

5.a - Se as i linhas de log formam um pacote maior que tamanho configurado, um pacote contendo i-1 linhas está pronto para ser transmitido e a i-ésima é inserida na íntegra em um novo pacote.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Execução em poucos milissegundos.

Frequência: Durante toda a sessão de utilização do *website*, a cada evento ocorrido no lado do cliente.

Canal até ator primário: Dados em memória.

Agendamento

Entrega: Versão 1.0

C.1.5 Envia log para o servidor

Informações características

Objetivo no contexto: Enviar dados armazenados em memória para o módulo servidor.

Escopo: Módulo cliente.

Nível: Tarefa primária.

Pré-condições: Pacotes estão gravados em memória.

Pós-condição de sucesso: Pacote é transmitido para o módulo servidor e o espaço que era ocupado pelo pacote é então liberado para novos pacotes.

Pós-condição de falha: Pacote deve ser retransmitido ou módulo cliente se encerra.

Ator primário: Módulo cliente.

Gatilho: Pacote é criado ou se 10 pacotes se acumulam na máquina do cliente.

Cenário principal de sucesso

1. Módulo cliente envia, de maneira assíncrona, os dados gravados para o módulo servidor, solicitando implicitamente o armazenamento dos pacotes.

2. Para cada pacote recebido o módulo servidor responde ao módulo cliente que a gravação foi bem sucedida.

3. Para cada resposta positiva vinda do servidor, o módulo cliente libera o espaço correspondente ao pacote gravado no servidor. A resposta do servidor deve conter o nome do pacote gravado, por exemplo, OK@[*nomedopacote*].

Extensões

3.a - Se a resposta do módulo servidor for negativa (e.g., $ERROR$@[*nomedopacote*]): módulo cliente não deve apagar o respectivo pacote, pois vai reenviá-lo em ciclo definido.

3.b - A cada ciclo de 10 segundos ou 10 pacotes acumulados no cliente: módulo cliente deve reenviar pacotes dos quais não recebeu resposta positiva.

3.c - Se a resposta do módulo servidor for $DENIED$@[*nomedopacote*], indicando que o *website* não está cadastrado, então o módulo cliente deve encerrar seu funcionamento.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Envio de *logs* e processamento das respostas em cerca de 2 segundos.

Frequência: Ao montar pacote ou se 10 ou mais pacotes estiverem acumulados.

Canal até ator primário: Dados em memória.

Atores secundários: Módulo servidor.

Canal até o ator secundário: HTTP(S).

Agendamento

Entrega: Versão 1.0.

C.1.6 Grava log

Informações características

Objetivo no contexto: Gravar os pacotes de *logs* no módulo servidor do WELFIT.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Módulo cliente envia pacote para o módulo servidor e módulo servidor está ativo e sem falhas.

Pós-condição de sucesso: Pacote é gravado e módulo servidor envia resposta para o módulo cliente.

Pós-condição de falha: Módulo servidor informa ao módulo cliente que houve erro na gravação dos *logs* enviados.

Ator primário: Módulo servidor.

Gatilho: Módulo servidor recebe dados do módulo cliente.

Cenário principal de sucesso

1. Módulo servidor recebe pacote do módulo cliente.

2. Módulo servidor verifica se o *website* ao qual o pacote se refere está cadastrado.

3. Módulo servidor grava o pacote e envia confirmação (e.g., *OK*@[*nomedopacote*]) para o módulo cliente, que pode então descartar o pacote que acabou de ser gravado.

Extensões

2.a - Se o *website* remetente não estiver cadastrado: módulo servidor envia a mensagem *DENIED*@[*nomedopacote*] para o módulo cliente, que então interrompe a captura no respectivo *website*.

3.a - Se houver algum erro na gravação: módulo servidor envia mensagem de erro (e.g., *ERROR*@[*nomedopacote*]) para o módulo cliente, que será responsável por reenviar o pacote que não foi gravado.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Gravação de pacote e envio das respostas em menos de 1 segundo.

Frequência: A cada segundo de utilização do *website* avaliado.

Canal até ator primário: Sistema de arquivos ou banco de dados.

Atores secundários: Módulo cliente.

Canal até o ator secundário: HTTP(S).

Agendamento

Entrega: Versão 1.0.

C.1.7 Cadastra avaliador

Informações características

Objetivo no contexto: Cadastrar avaliador.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: E-mail do novo avaliador não está cadastrado no sistema.

Pós-condição de sucesso: Cadastro do novo avaliador está validado no sistema.

Pós-condição de falha: Cadastro não é concluído.

Ator primário: Avaliador de *websites*.

Gatilho: Avaliador acessa a página de cadastro.

Cenário principal de sucesso

1. Novo avaliador preenche nome, e-mail e senha.

2. Sistema verifica se e-mail não está cadastrado e validado.

3. Novo avaliador recebe a confirmação do cadastro.

Informações relacionadas

Prioridade: Média.

Meta de performance: Menos de 1 minuto.

Frequência: Uma vez para cada avaliador cadastrado.

Canal até ator primário: Página de cadastro e e-mail de validação de e-mail.

Agendamento

Entrega: Versão 1.0.

C.1.8 Autenticação

Informações características

Objetivo no contexto: Autenticar no sistema.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Avaliador está cadastrado.

Pós-condição de sucesso: Avaliador está autenticado para utilizar o sistema.

Pós-condição de falha: Avaliador não pode acessar áreas restritas do sistema.

Ator primário: Avaliador.

Gatilho: Acessar qualquer página de áreas restritas do sistema.

Cenário principal de sucesso

1. Avaliador preenche e-mail e senha e envia o formulário de autenticação.

2. Sistema devolve para o avaliador autenticado a página inicial da administração da ferramenta contendo os *websites* associados a ele, as opções para editar informações de cada *website* (e.g., restringir a captura de *logs* em regiões do *website*) e solicitar relatórios.

Extensões

2.a - Avaliador não está cadastrado no sistema ou senha incorreta: Sistema devolve para o avaliador uma mensagem informando o ocorrido.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 10 segundos.

Frequência: Uma vez por sessão para cada avaliador que acessar o sistema.

Canal até ator primário: Página de autenticação do sistema.

Agendamento

Entrega: Versão 1.0.

C.1.9 Atualiza avaliador

Informações características

Objetivo no contexto: Atualizar informações de avaliadores da ferramenta.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Avaliador está autenticado.

Pós-condição de sucesso: Dados do avaliador estão atualizados.

Pós-condição de falha: Dados do avaliador não sofrem alterações.

Ator primário: Avaliador.

Gatilho: Avaliador acessa página de edição de informações de avaliador.

Cenário principal de sucesso

1. Avaliador altera suas informações (e.g., nome, e-mail, senha).
2. Avaliador confirma alterações.
3. Sistema valida informações e apresenta mensagem de confirmação para o avaliador.

Informações relacionadas

Prioridade: Média.

Meta de performance: Menos de 1 minuto.

Frequência: Não é possível especificar.

Canal até ator primário: Página de edição de informações de avaliadores.

Agendamento

Entrega: Versão 1.0.

C.1.10 Cadastra *website*

Informações características

Objetivo no contexto: Cadastrar *websites* para que o WELFIT os avalie.
Escopo: Módulo servidor.
Nível: Tarefa primária.
Pré-condições: Avaliador está autenticado e *website* não está cadastrado.
Pós-condição de sucesso: *Website* está cadastrado.
Pós-condição de falha: *Website* não foi cadastrado.
Ator primário: Avaliador.
Gatilho: Avaliador acessa página de cadastro de *websites*.

Cenário principal de sucesso

1. Avaliador preenche informações (Nome, URL do nível mais abrangente e expressão regular para bloquear a captura de *logs* em algumas áreas do *website* avaliado, por exemplo, */login.html*) e confirma.
2. Sistema verifica informações preenchidas.
3. Sistema retorna mensagem de confirmação para o avaliador.

Informações relacionadas

Prioridade: Alta.
Meta de performance: Menos de 1 minuto.
Frequência: Aproximadamente uma vez para cada avaliador cadastrado.
Canal até ator primário: Página de cadastro de *websites*.

Agendamento

Entrega: Versão 1.0.

C.1.11 Atualiza *website*

Informações características

Objetivo no contexto: Atualizar dados de *websites* cadastrados no sistema.
Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Avaliador está autenticado e *website* está cadastrado.

Pós-condição de sucesso: Dados do *website* estão atualizados.

Pós-condição de falha: Dados do *website* não são alterados.

Ator primário: Avaliador.

Gatilho: Avaliador acessa página de edição de informações de *website*.

Cenário principal de sucesso

1. Avaliador altera informações (Nome, URL do nível mais abrangente e expressão regular para bloquear a captura de *logs* em algumas áreas do *website* avaliado, por exemplo, */login.html*) e confirma.

2. Sistema verifica informações preenchidas.

3. Sistema retorna mensagem de confirmação para o avaliador.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 1 minuto.

Frequência: Não é possível especificar.

Canal até ator primário: Página de edição de *website*.

Agendamento

Entrega: Versão 1.0.

C.1.12 Solicita relatório

Informações características

Objetivo no contexto: Solicitar relatório de avaliação.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Avaliador está autenticado e existe algum *website* associado ao avaliador solicitante.

Pós-condição de sucesso: Avaliador recebe relatório.

Pós-condição de falha: Avaliador é notificado da falha ocorrida.

Ator primário: Avaliador.

Gatilho: Avaliador acessa página da ferramenta e solicita relatório de *websites*.

Cenário principal de sucesso

1. Avaliador seleciona *website*.

2. Avaliador solicita relatório.

3. Sistema retorna relatório para avaliador.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 1 minuto.

Frequência: Não é possível especificar.

Canal até ator primário: Página Web de solicitação de relatório.

Agendamento

Entrega: Versão 1.1.

C.1.13 Gera relatório

Informações características

Objetivo no contexto: Montar relatório para o avaliador que o solicitou.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Relatório foi solicitado por avaliador autenticado, distância entre sessões está calculada e *cluster* foi definido.

Pós-condição de sucesso: Relatório apresentado para o avaliador que o solicitou.

Pós-condição de falha: Avaliador é notificado de que ocorreu alguma falha.

Ator primário: Módulo servidor.

Gatilho: Avaliador solicita relatório de *website* administrado por ele.

Cenário principal de sucesso

1. Módulo servidor calcula distância entre sessões relativas ao relatório solicitado.

2. Módulo servidor agrupa os fluxos de eventos (i.e., identifica barreiras que os participantes enfrentaram, tendo foco os fluxos relacionados aos participantes que utilizam tecnologias assistivas).

3. Módulo servidor gera os grafos com os fluxos de utilização e destacam as barreiras identificadas na fase anterior.

4. Módulo servidor reúne todas as informações do *website* relacionado à solicitação e apresenta o relatório para o avaliador solicitante.

Informações relacionadas

Prioridade: Alta.
Meta de performance: Montar relatórios semanais em menos de 1 minuto.
Frequência: Não é possível especificar.
Canal até ator primário: Sistema de arquivos ou bancos de dados.

Agendamento

Entrega: Versão 1.1.

C.1.14 Calcula distância entre sessões

Informações características

Objetivo no contexto: Calcula distância entre as sessões de uma determinada URL.
Escopo: Módulo servidor.
Nível: Tarefa primária.
Pré-condições: Novas sessões gravadas no módulo servidor.
Pós-condição de sucesso: Distância entre sessões estão gravadas no banco de dados.
Pós-condição de falha: Distância entre sessões não foram calculadas/gravadas.
Ator primário: Módulo servidor.
Gatilho: Avaliador solicita relatório ou em ciclo definido.

Cenário principal de sucesso

1. Novas sessões têm sua distância calculada com todas as outras sessões em uma dada URL.

Informações relacionadas

Prioridade: Alta.
Meta de performance: Poucos segundos por sessão.
Frequência: A cada 30 minutos.
Canal até ator primário: Bancos de dados.

Agendamento

Entrega: Versão 1.1.

C.1.15 Agrupa *logs*

Informações características

Objetivo no contexto: Agrupar *logs* de acordo com a distância entre as sessões.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Distância entre as sessões estão calculadas.

Pós-condição de sucesso: Grupos são definidos e gravados em banco de dados de forma que possibilitem a geração de grafos das cadeias de eventos relativas a essas sessões.

Pós-condição de falha: Sessões não são agrupadas.

Ator primário: Módulo servidor.

Gatilho: Avaliador acessa página e solicita geração de grupos ou assim que distância entre novas sessões são calculadas.

Cenário principal de sucesso

1. Cadeias de eventos são agrupadas considerando algoritmo hierárquico aglomerativo considerando o average linkage na fusão de grupos. Note que o número ótimo de grupos deve ser identificado pelo algoritmo.

2. Grupos são gravados em banco de dados.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 10 segundos.

Frequência: A cada 30 minutos.

Canal até ator primário: Banco de dados.

Agendamento

Entrega: Versão 2.0.

C.1.16 Desenha grafo

Informações características

Objetivo no contexto: Desenhar o grafo de utilização para que especialistas possam analisar as barreiras identificadas.

Escopo: Módulo servidor.

Nível: Sub-função.

Pré-condições: Representação do grafo de fluxos agrupados e classificados está disponível no sistema e programa externo para desenhar o grafo está acessível para o sistema.

Pós-condição de sucesso: Grafo com os fluxos de utilização está disponível para o avaliador.

Pós-condição de falha: Grafo com fluxos de utilização não foi gerado.

Ator primário: Módulo servidor.

Gatilho: Avaliador solicita relatório.

Cenário principal de sucesso

1. Módulo servidor recupera sessões relativas ao agrupamento.

2. Módulo servidor recupera linhas de log relativas às sessões recuperadas.

3. Módulo servidor utiliza programa externo e gera grafo representando os fluxos de utilização tendo destacadas as barreiras enfrentadas pelos participantes. No grafo gerado os nós são os eventos.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 10 segundos por grafo.

Frequência: Não é possível definir.

Canal até ator primário: Bancos de dados.

Atores secundários: Programa externo para geração de grafos (e.g., JGraph).

Canal até o ator secundário: Biblioteca de descrição de grafo.

Agendamento

Entrega: Versão 2.0.

C.1.17 Cadastra norma

Informações características

Objetivo no contexto: Cadastrar normas, envolvendo o seletor de determinado trecho na cadeia de eventos.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Avaliador está autenticado e é administrador do WELFIT.

Pós-condição de sucesso: Norma é cadastrada com sucesso.

Pós-condição de falha: Norma não é cadastrada.

Ator primário: Administrador do WELFIT.

Gatilho: Administrador do WELFIT acessa página de cadastro de normas.

Cenário principal de sucesso

1. Administrador do WELFIT preenche campos de descrição, seletor de cadeia de eventos e os ajustes a serem aplicados.

Cenário principal de sucesso

1a. Administrador do WELFIT pode testar seletor e ajustes a partir de uma dada URL que esteja em avaliação e que conte com eventos gravados no banco de dados.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 10 minutos.

Frequência: Não é possível prever.

Canal até ator primário: Banco de dados.

Agendamento

Entrega: Versão 2.0.

C.1.18 Inicia experimento

Informações características

Objetivo no contexto: Selecionar ajustes a serem aplicados para determinados *clusters*.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: *Clusters* estão definidos, ao menos uma norma deve estar cadastrada e não há nenhum experimento rodando para o mesmo *cluster*-url.

Pós-condição de sucesso: Experimento é iniciado marcando data de início e ajuste a ser aplicado para determinado *cluster* de certa URL.

Pós-condição de falha: Experimento não é iniciado.

Ator primário: Módulo servidor.

Gatilho: Intervalo de tempo de criação do último experimento (e.g., a cada 7 dias).

Cenário principal de sucesso

1. Selecionar todos os *clusters* que se aplicam para cada norma e registrar o ajuste concreto com base na memória da máquina de estados. Os dados são gravados em banco sob o status de experimento em elaboração. Neste momento o número de incidentes de uso pré-teste está pendente.

2. Para cada *cluster* o número de incidentes de uso é calculado e gravado em banco.

3. Timestamp é atualizado e status do experimento é alterado para em execução.

Extensões

3.a – Se o experimento iniciado é uma nova etapa de um experimento existente, isto é, se havia um experimento para o mesmo *cluster*, mesma URL e mesma norma, então, se o número de incidentes de uso reduziu, então o contador de sucessos na tabela de normas é incrementado para a norma aplicada no experimento.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 10 minutos.

Frequência: Uma vez por semana.

Canal até ator primário: Banco de dados.

Agendamento

Entrega: Versão 2.0.

C.1.19 Recuperar ajustes

Informações características

Objetivo no contexto: Selecionar ajustes a serem aplicados para um participante que já respondeu ao convite.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Módulo cliente enviou ticket e URL em que o participante está e há algum experimento em curso.

Pós-condição de sucesso: Ajustes para um determinado ticket e URL são selecionados.

Pós-condição de falha: Nenhum ajuste é recuperado.

Ator primário: Módulo servidor.

Gatilho: Módulo cliente identifica que o dispositivo do cliente conta com um cookie contendo o ticket.

Cenário principal de sucesso

1. Módulo servidor seleciona os ids de sessão para o ticket e URL informados.

2. A partir do id de sessão mais recente do ticket e URL informados, o módulo servidor seleciona *cluster*.

3. A partir do *cluster* e URL selecionados, módulo servidor seleciona experimentos ao qual este *cluster* está participando.

4. Uma vez selecionados os experimentos, são recuperadas as normas e, consequentemente, os ajustes a serem aplicados para a URL que um determinado ticket antigo está revisitando.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 5 segundos.

Frequência: A partir da segunda visita de um participante em uma URL sendo avaliada.

Canal até ator primário: Banco de dados.

Agendamento

Entrega: Versão 2.0.

C.1.20 Ajusta IU

Informações características

Objetivo no contexto: Ajustar a Interface de Usuário de forma a reduzir números de problemas identificados no grafo de uso gerado com base nas cadeias de eventos.

Escopo: Módulo cliente.

Nível: Tarefa primária.

Pré-condições: Ajustes foram recuperados no servidor e enviados para o módulo cliente.

Pós-condição de sucesso: IU é ajustada.

Pós-condição de falha: IU não é ajustada.

Ator primário: Módulo cliente.

Gatilho: Módulo servidor envia ajustes a serem aplicados.

Cenário principal de sucesso

1. Módulo cliente identifica que o dispositivo do cliente conta com um ticket antigo.

2. Módulo cliente envia ticket e URL sendo visitada como parte da requisição de ajustes a serem aplicados.

3. Módulo servidor responde requisição contendo os ajustes a serem aplicados

4. Módulo cliente manipula a árvore DOM para ajustar a IU.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 10 segundos para aplicar as alterações.

Frequência: Assim que um participante visita uma página pela segunda vez.

Canal até ator primário: Bancos de dados e requisição AJAX.

Agendamento

Entrega: Versão 2.0.

C.1.21 Encerra experimentos

Informações características

Objetivo no contexto: Encerrar experimentos que contam com visitas.

Escopo: Módulo servidor.

Nível: Tarefa primária.

Pré-condições: Experimentos estão criados e contam com visitas nos grupos experimental e controle.

Pós-condição de sucesso: Experimento encerrado marcando se foi bem sucedido ou mal sucedido.

Pós-condição de falha: Experimento não é encerrado.

Ator primário: Módulo servidor.

Gatilho: Intervalo de tempo de encerramento do último experimento (e.g., a cada 7 dias).

Cenário principal de sucesso

1. Módulo servidor seleciona tickets dos participantes que retornaram após terem sido sorteados como parte do grupo experimental dos experimentos criados.

2. Grafos de uso são montados (em memória) para cada url-*cluster*-grupo são montados para ocorrer a contabilização dos incidentes de uso e posterior comparação entre experimental e controle.

3. Se o número de incidentes de uso do grupo experimental, ponderado pelo número de sessões do grupo, é menor do que o número de incidentes do grupo de controle, ponderado pelo número de sessões do grupo, então o experimento tem seu status alterado para bem sucedido; caso contrário o status é alterado para mal sucedido.

Informações relacionadas

Prioridade: Alta.

Meta de performance: Menos de 10 minutos.

Frequência: Uma vez por semana.

Canal até ator primário: Banco de dados.

Agendamento

Entrega: Versão 2.0.

C.2 Diagrama de casos de uso

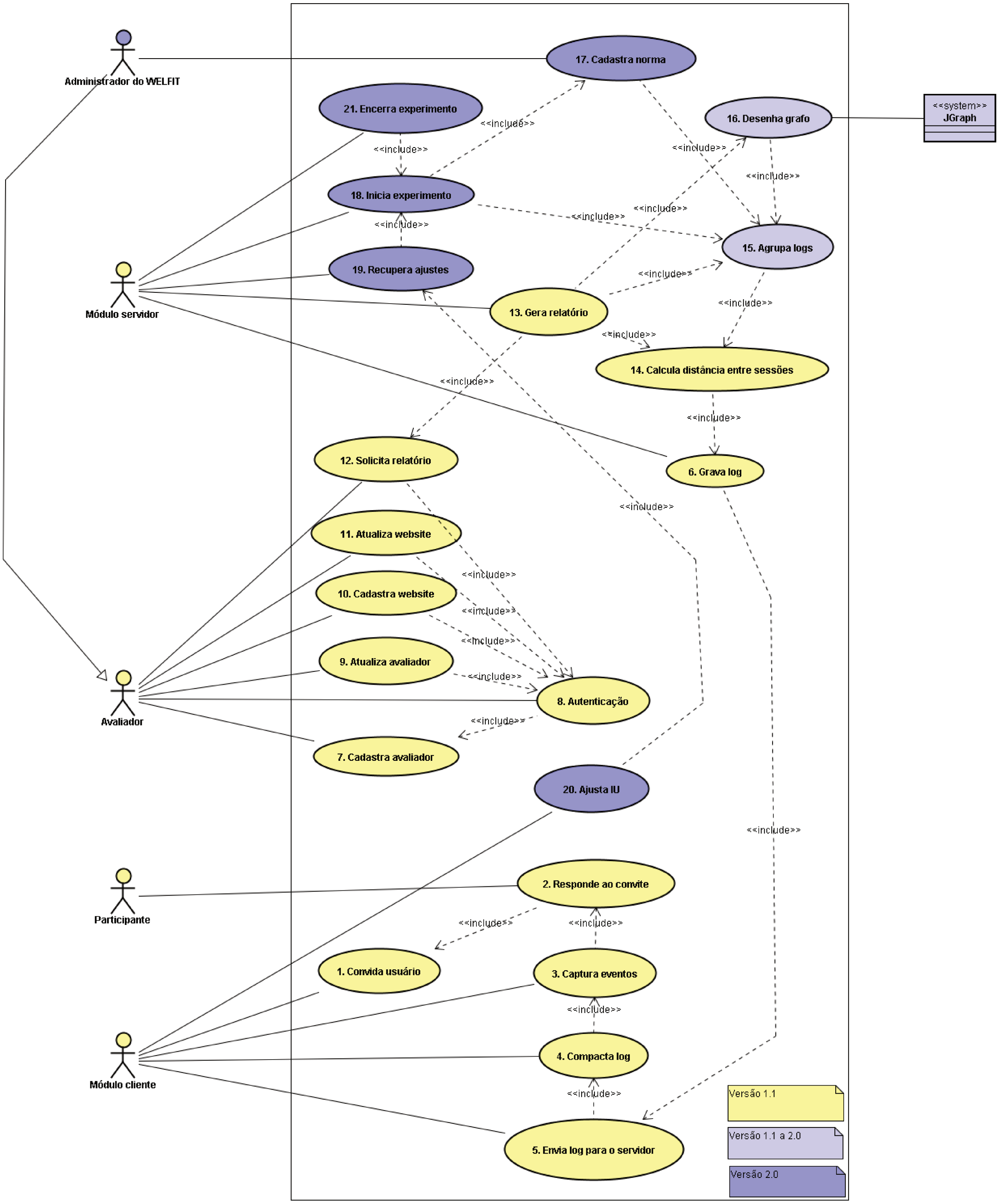


Figure C.1: Diagrama de casos de uso.

C.3 Banco de dados

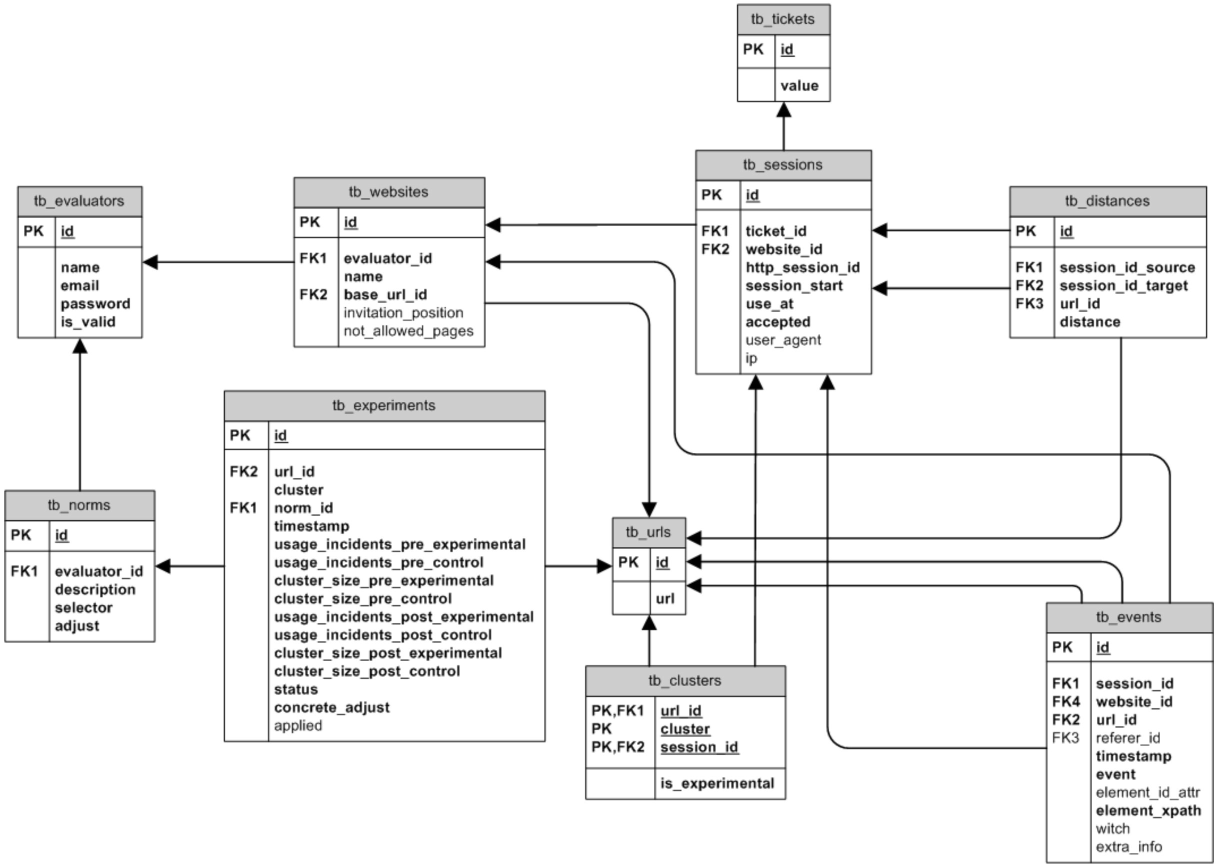


Figure C.2: Modelo de banco de dados

Rationale do uso de IDs: o nome de eventos não foi usado como ID, pois eventos podem ser customizados e por conta do HTML5 vão mudar. O mesmo se aplica ao nome das tags; para as URLs foi usado ID, pois para a seleção de *clusters*, se usar *strings*, a consulta, que envolverá centenas/milhares de registros, ficaria significantemente mais lenta.

C.4 Fluxograma

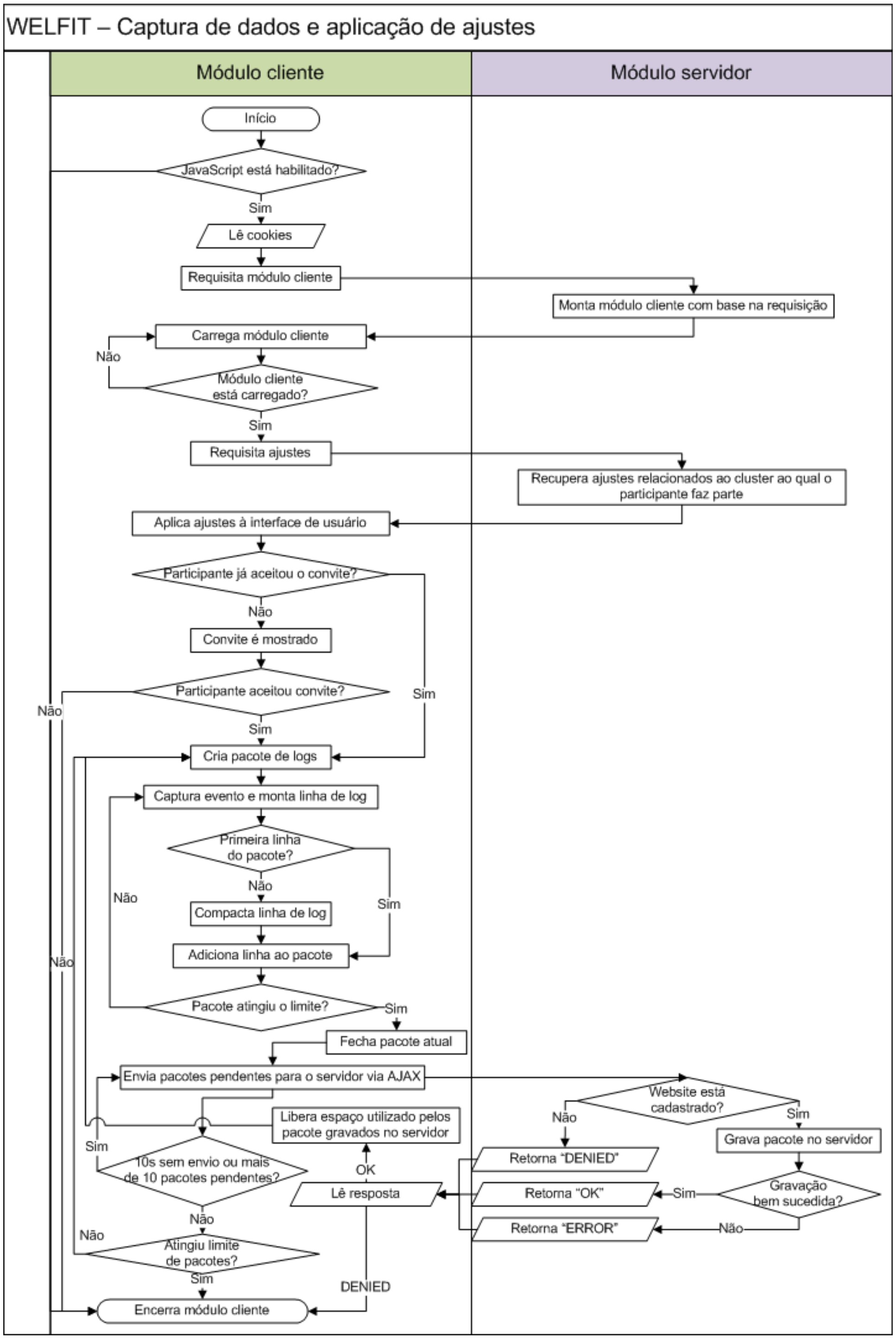
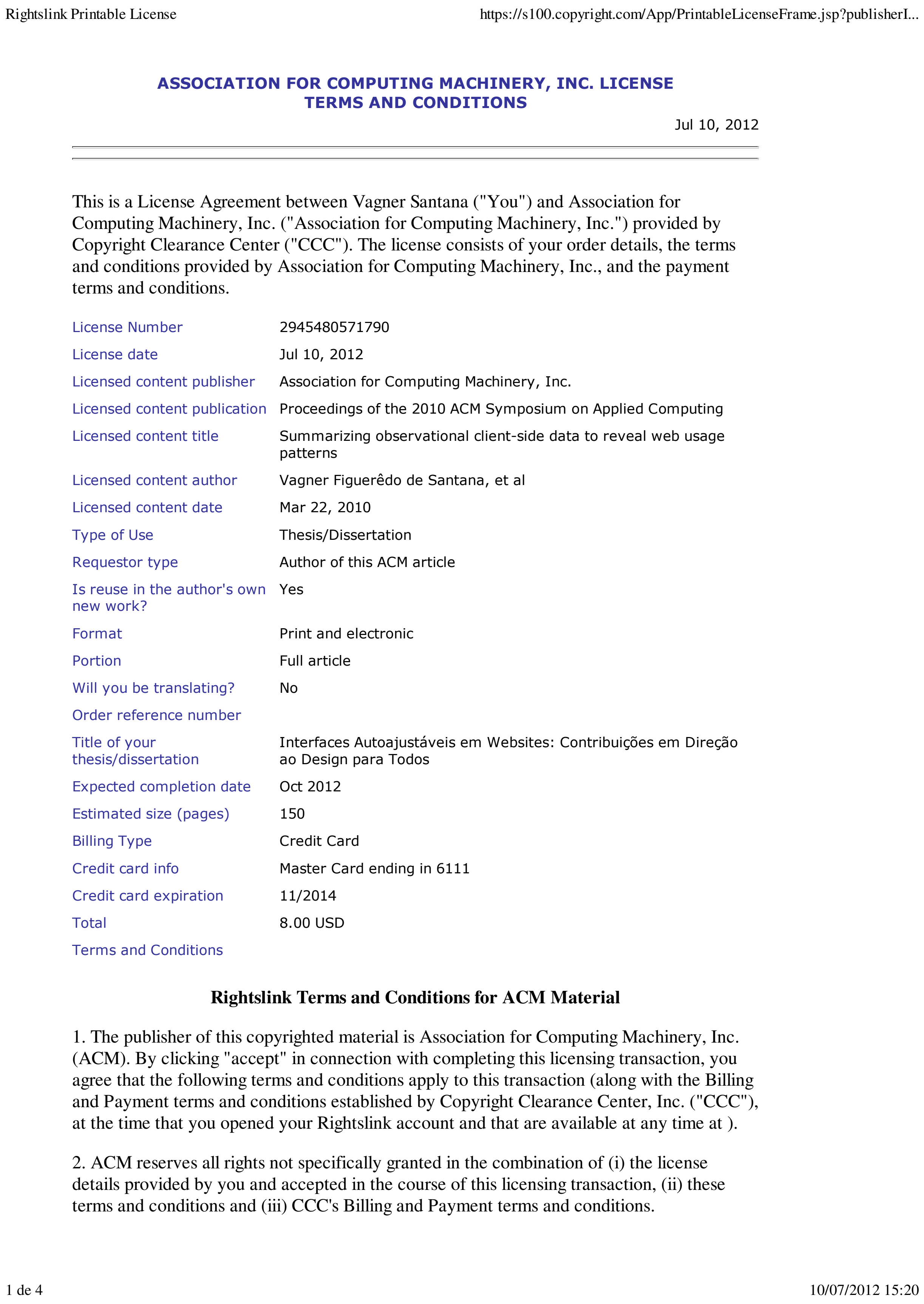

WELFIT – Captura de dados e aplicação de ajustes
Módulo cliente
Módulo servidor
Início
JavaScript está habilitado?
Sim
Lê cookies
Requisita módulo cliente
Monta módulo cliente com base na requisição
Carrega módulo cliente
Não
Módulo cliente está carregado?
Sim
Requisita ajustes
Recupera ajustes relacionados ao cluster ao qual o participante faz parte
Aplica ajustes à interface de usuário
Participante já aceitou o convite?
Não
Convite é mostrado
Participante aceitou convite?
Não
Sim
Sim
Cria pacote de logs
Captura evento e monta linha de log
Primeira linha do pacote?
Não
Sim
Não
Compacta linha de log
Adiciona linha ao pacote
Não
Pacote atingiu o limite?
Sim
Fecha pacote atual
Envia pacotes pendentes para o servidor via AJAX
Website está cadastrado?
Não
Sim
Libera espaço utilizado pelos pacote gravados no servidor
Grava pacote no servidor
Sim
10s sem envio ou mais de 10 pacotes pendentes?
OK
Retorna “DENIED”
Lê resposta
Retorna “OK”
Sim
Gravação bem sucedida?
Retorna “ERROR”
Não
Não
Não
Atingiu limite de pacotes?
Sim
DENIED
Encerra módulo cliente

Figure C.3: Fluxograma da captura de *logs* recuperaçao de ajustes

Appendix D

Autorizações para utilização de artigos



ASSOCIATION FOR COMPUTING MACHINERY, INC. LICENSE TERMS AND CONDITIONS

Jul 10, 2012

This is a License Agreement between Vagner Santana ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Association for Computing Machinery, Inc., and the payment terms and conditions.

License Number	2945480571790
License date	Jul 10, 2012
Licensed content publisher	Association for Computing Machinery, Inc.
Licensed content publication	Proceedings of the 2010 ACM Symposium on Applied Computing
Licensed content title	Summarizing observational client-side data to reveal web usage patterns
Licensed content author	Vagner Figuerêdo de Santana, et al
Licensed content date	Mar 22, 2010
Type of Use	Thesis/Dissertation
Requestor type	Author of this ACM article
Is reuse in the author's own new work?	Yes
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Order reference number	
Title of your thesis/dissertation	Interfaces Autoajustáveis em Websites: Contribuições em Direção ao Design para Todos
Expected completion date	Oct 2012
Estimated size (pages)	150
Billing Type	Credit Card
Credit card info	Master Card ending in 6111
Credit card expiration	11/2014
Total	8.00 USD
Terms and Conditions	

Rightslink Terms and Conditions for ACM Material

1. The publisher of this copyrighted material is Association for Computing Machinery, Inc. (ACM). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at).

2. ACM reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

3. ACM hereby grants to licensee a non-exclusive license to use or republish this ACM-copyrighted material* in secondary works (especially for commercial distribution) with the stipulation that consent of the lead author has been obtained independently. Unless otherwise stipulated in a license, grants are for one-time use in a single edition of the work, only with a maximum distribution equal to the number that you identified in the licensing process. Any additional form of republication must be specified according to the terms included at the time of licensing.

*Please note that ACM cannot grant republication or distribution licenses for embedded third-party material. You must confirm the ownership of figures, drawings and artwork prior to use.

4. Any form of republication or redistribution must be used within 180 days from the date stated on the license and any electronic posting is limited to a period of six months unless an extended term is selected during the licensing process. Separate subsidiary and subsequent republication licenses must be purchased to redistribute copyrighted material on an extranet. These licenses may be exercised anywhere in the world.

5. Licensee may not alter or modify the material in any manner (except that you may use, within the scope of the license granted, one or more excerpts from the copyrighted material, provided that the process of excerpting does not alter the meaning of the material or in any way reflect negatively on the publisher or any writer of the material).

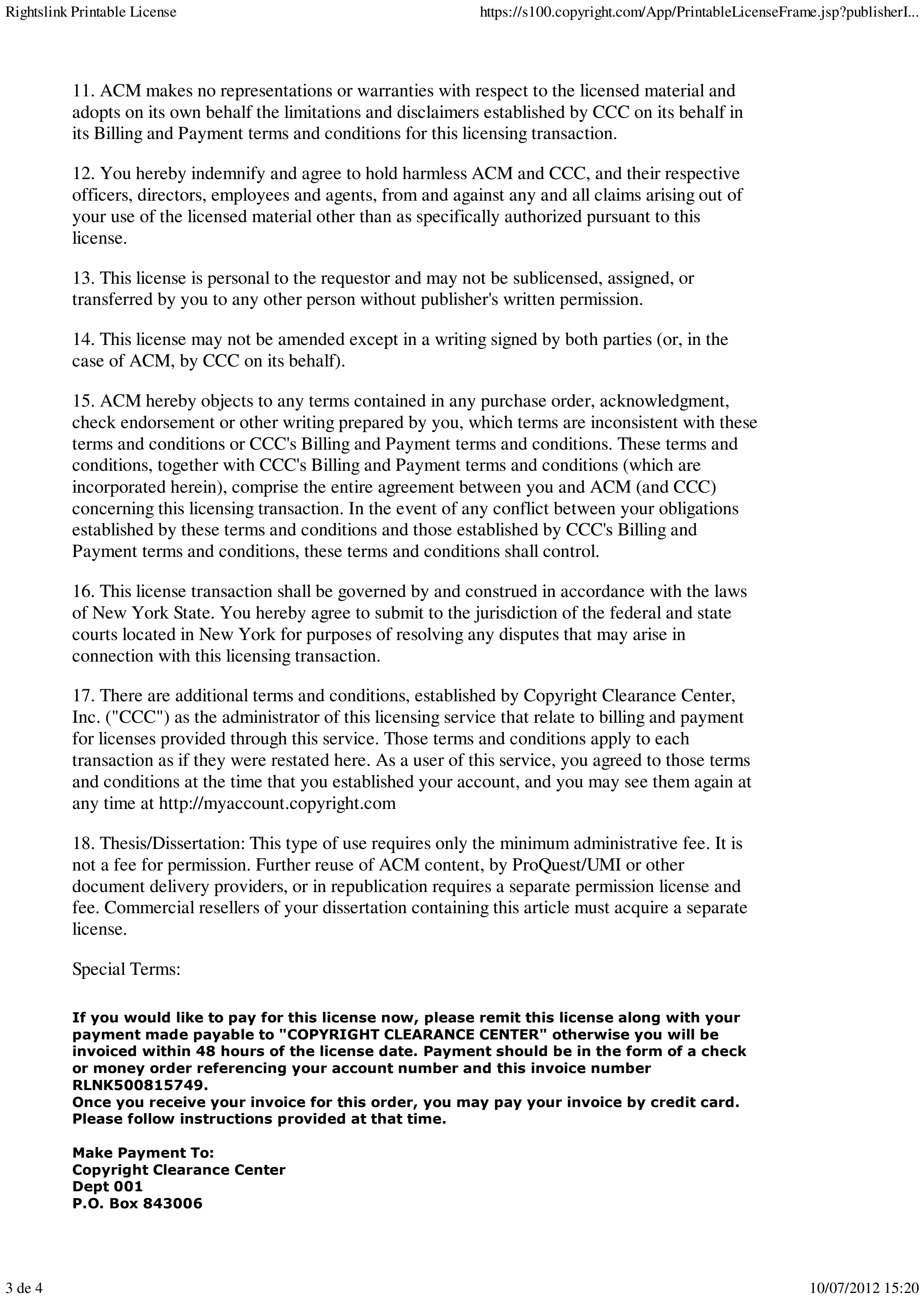
6. Licensee must include the following copyright and permission notice in connection with any reproduction of the licensed material: "[Citation] © YEAR Association for Computing Machinery, Inc. Reprinted by permission." Include the article DOI as a link to the definitive version in the ACM Digital Library. Example: Charles, L. "How to Improve Digital Rights Management," Communications of the ACM, Vol. 51:12, © 2008 ACM, Inc. http://doi.acm.org/10.1145/nnnnnn.nnnnnn (where nnnnnn.nnnnnn is replaced by the actual number).

7. Translation of the material in any language requires an explicit license identified during the licensing process. Due to the error-prone nature of language translations, Licensee must include the following copyright and permission notice and disclaimer in connection with any reproduction of the licensed material in translation: "This translation is a derivative of ACM-copyrighted material. ACM did not prepare this translation and does not guarantee that it is an accurate copy of the originally published work. The original intellectual property contained in this work remains the property of ACM."

8. You may exercise the rights licensed immediately upon issuance of the license at the end of the licensing transaction, provided that you have disclosed complete and accurate details of your proposed use. No license is finally effective unless and until full payment is received from you (either by CCC or ACM) as provided in CCC's Billing and Payment terms and conditions.

9. If full payment is not received within 90 days from the grant of license transaction, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted.

10. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.



11. ACM makes no representations or warranties with respect to the licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

12. You hereby indemnify and agree to hold harmless ACM and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

13. This license is personal to the requestor and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

14. This license may not be amended except in a writing signed by both parties (or, in the case of ACM, by CCC on its behalf).

15. ACM hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and ACM (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

16. This license transaction shall be governed by and construed in accordance with the laws of New York State. You hereby agree to submit to the jurisdiction of the federal and state courts located in New York for purposes of resolving any disputes that may arise in connection with this licensing transaction.

17. There are additional terms and conditions, established by Copyright Clearance Center, Inc. ("CCC") as the administrator of this licensing service that relate to billing and payment for licenses provided through this service. Those terms and conditions apply to each transaction as if they were restated here. As a user of this service, you agreed to those terms and conditions at the time that you established your account, and you may see them again at any time at http://myaccount.copyright.com

18. Thesis/Dissertation: This type of use requires only the minimum administrative fee. It is not a fee for permission. Further reuse of ACM content, by ProQuest/UMI or other document delivery providers, or in republication requires a separate permission license and fee. Commercial resellers of your dissertation containing this article must acquire a separate license.

Special Terms:

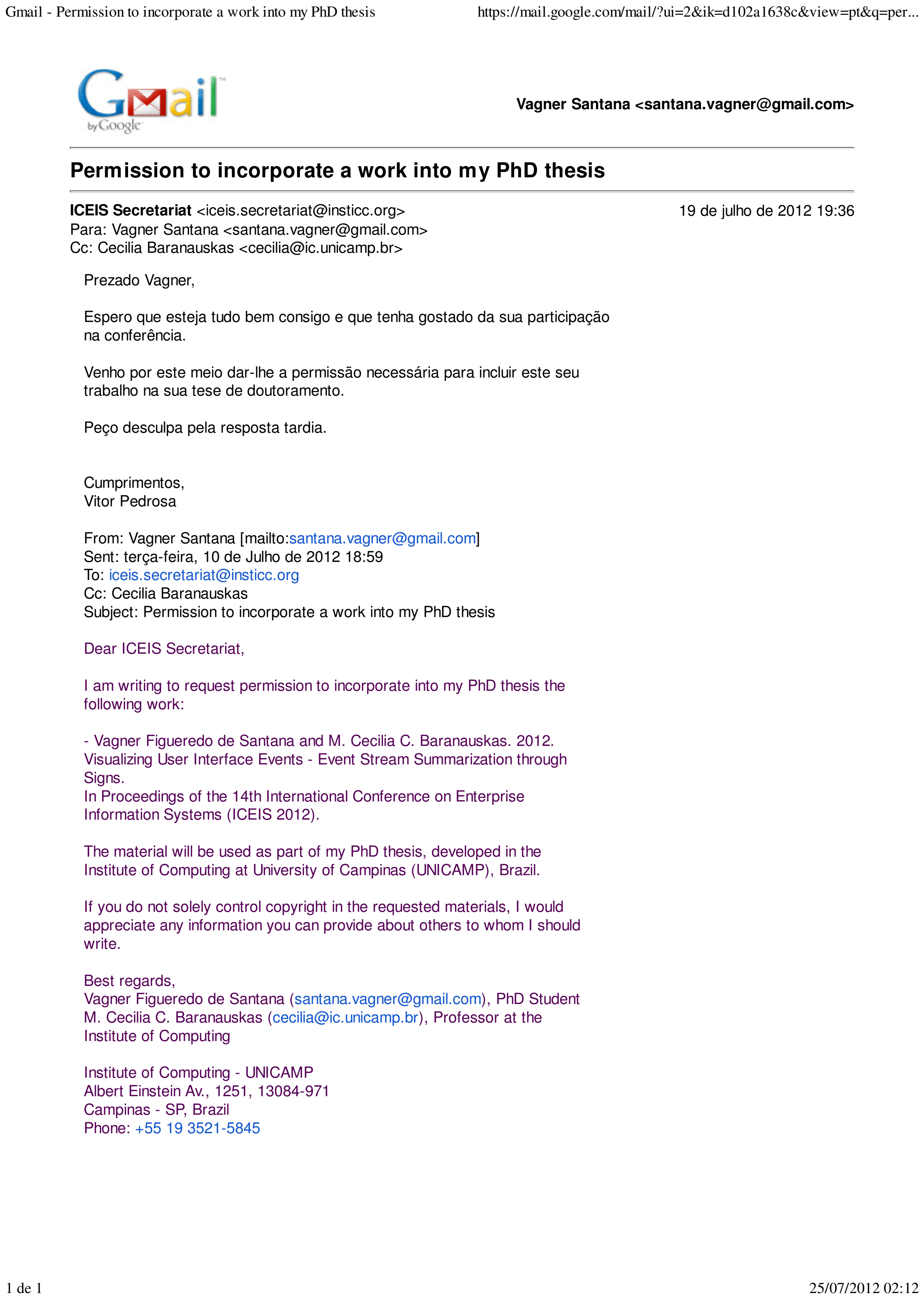
If you would like to pay for this license now, please remit this license along with your payment made payable to "COPYRIGHT CLEARANCE CENTER" otherwise you will be invoiced within 48 hours of the license date. Payment should be in the form of a check or money order referencing your account number and this invoice number RLNK500815749.
Once you receive your invoice for this order, you may pay your invoice by credit card. Please follow instructions provided at that time.

Make Payment To:
Copyright Clearance Center
Dept 001
P.O. Box 843006

Boston, MA 02284-3006

For suggestions or comments regarding this order, contact RightsLink Customer Support: customercare@copyright.com or +1-877-622-5543 (toll free in the US) or +1-978-646-2777.

Gratis licenses (referencing $0 in the Total field) are free. Please retain this printable license for your reference. No payment is required.



Gmail - Permission to incorporate a work into my PhD thesis https://mail.google.com/mail/?ui=2&ik=d102a1638c&view=pt&q=per...

Vagner Santana <santana.vagner@gmail.com>

Permission to incorporate a work into my PhD thesis

ICEIS Secretariat <iceis.secretariat@insticc.org> 19 de julho de 2012 19:36
Para: Vagner Santana <santana.vagner@gmail.com>
Cc: Cecilia Baranauskas <cecilia@ic.unicamp.br>

Prezado Vagner,

Espero que esteja tudo bem consigo e que tenha gostado da sua participação na conferência.

Venho por este meio dar-lhe a permissão necessária para incluir este seu trabalho na sua tese de doutoramento.

Peço desculpa pela resposta tardia.

Cumprimentos,
Vitor Pedrosa

From: Vagner Santana [mailto:santana.vagner@gmail.com]
Sent: terça-feira, 10 de Julho de 2012 18:59
To: iceis.secretariat@insticc.org
Cc: Cecilia Baranauskas
Subject: Permission to incorporate a work into my PhD thesis

Dear ICEIS Secretariat,

I am writing to request permission to incorporate into my PhD thesis the following work:

- Vagner Figueredo de Santana and M. Cecilia C. Baranauskas. 2012. Visualizing User Interface Events - Event Stream Summarization through Signs.
In Proceedings of the 14th International Conference on Enterprise Information Systems (ICEIS 2012).

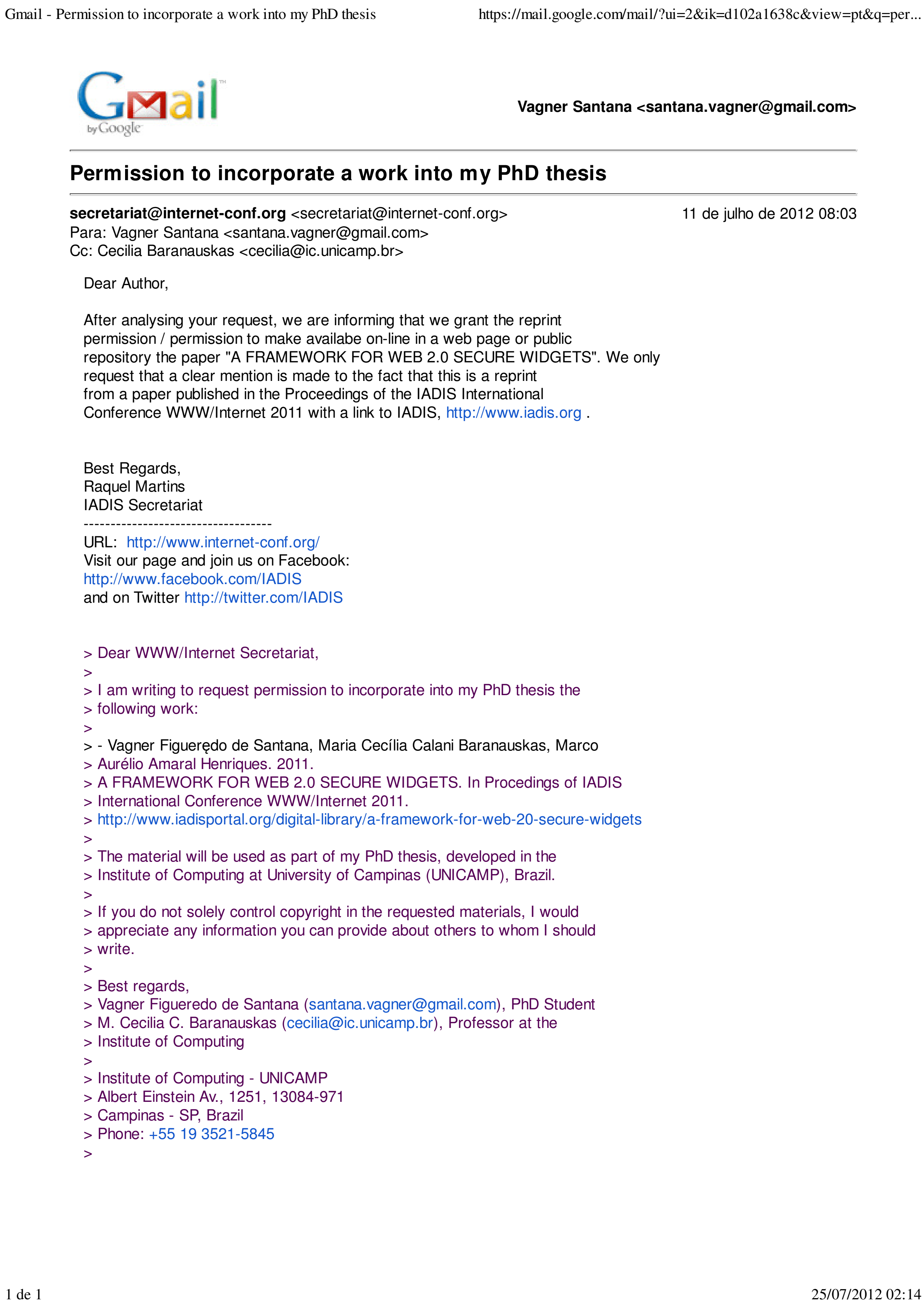
The material will be used as part of my PhD thesis, developed in the Institute of Computing at University of Campinas (UNICAMP), Brazil.

If you do not solely control copyright in the requested materials, I would appreciate any information you can provide about others to whom I should write.

Best regards,
Vagner Figueredo de Santana (santana.vagner@gmail.com), PhD Student
M. Cecilia C. Baranauskas (cecilia@ic.unicamp.br), Professor at the Institute of Computing

Institute of Computing - UNICAMP
Albert Einstein Av., 1251, 13084-971
Campinas - SP, Brazil
Phone: +55 19 3521-5845

1 de 1 25/07/2012 02:12



Gmail - Permission to incorporate a work into my PhD thesis https://mail.google.com/mail/?ui=2&ik=d102a1638c&view=pt&q=per...

Gmail by Google

Vagner Santana <santana.vagner@gmail.com>

Permission to incorporate a work into my PhD thesis

secretariat@internet-conf.org <secretariat@internet-conf.org> 11 de julho de 2012 08:03
Para: Vagner Santana <santana.vagner@gmail.com>
Cc: Cecilia Baranauskas <cecilia@ic.unicamp.br>

Dear Author,

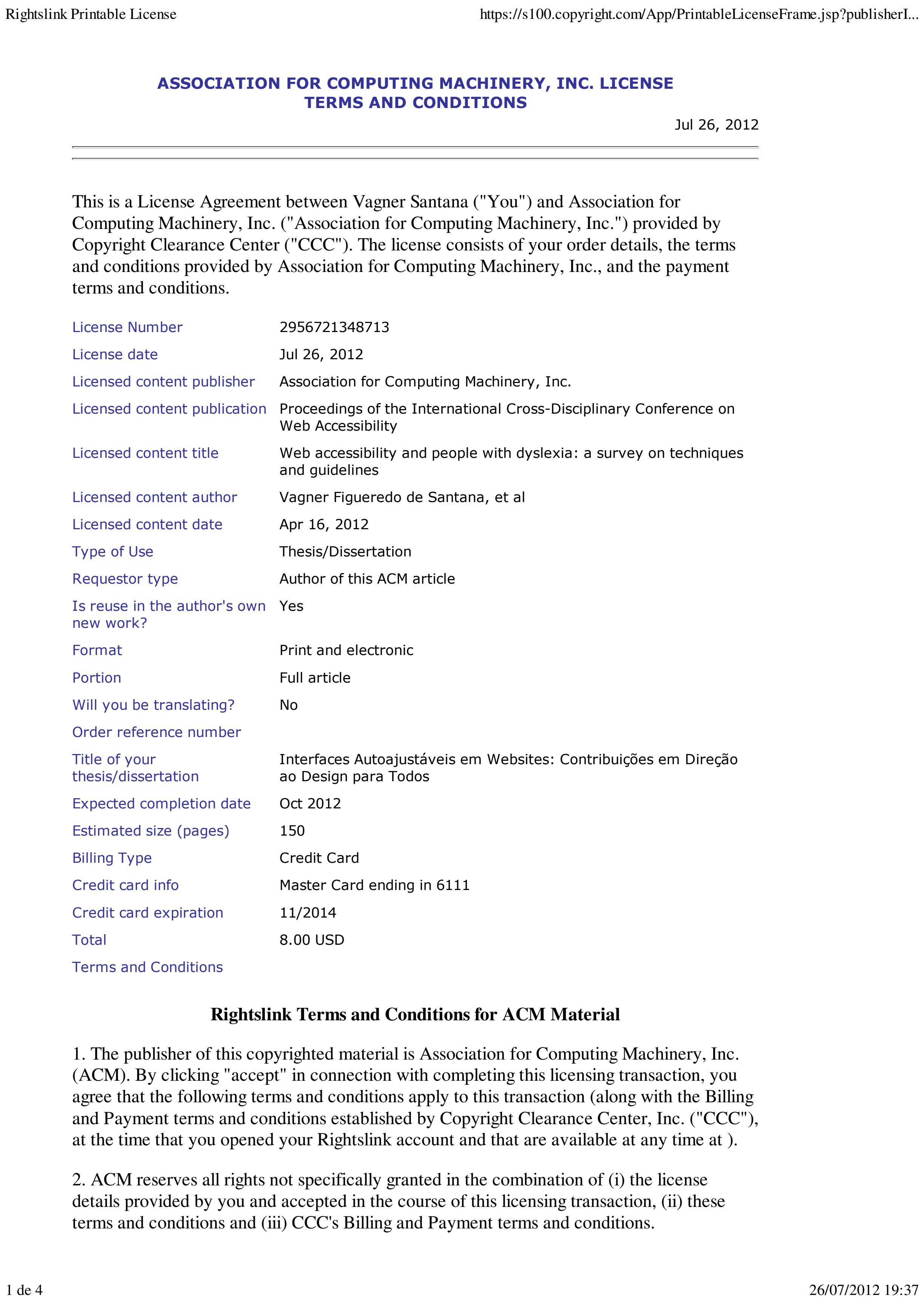
After analysing your request, we are informing that we grant the reprint permission / permission to make availabe on-line in a web page or public repository the paper "A FRAMEWORK FOR WEB 2.0 SECURE WIDGETS". We only request that a clear mention is made to the fact that this is a reprint from a paper published in the Proceedings of the IADIS International Conference WWW/Internet 2011 with a link to IADIS, http://www.iadis.org .

Best Regards,
Raquel Martins
IADIS Secretariat

URL: http://www.internet-conf.org/
Visit our page and join us on Facebook:
http://www.facebook.com/IADIS
and on Twitter http://twitter.com/IADIS

> Dear WWW/Internet Secretariat,
>
> I am writing to request permission to incorporate into my PhD thesis the
> following work:
>
> - Vagner Figuerędo de Santana, Maria Cecília Calani Baranauskas, Marco
> Aurélio Amaral Henriques. 2011.
> A FRAMEWORK FOR WEB 2.0 SECURE WIDGETS. In Procedings of IADIS
> International Conference WWW/Internet 2011.
> http://www.iadisportal.org/digital-library/a-framework-for-web-20-secure-widgets
>
> The material will be used as part of my PhD thesis, developed in the
> Institute of Computing at University of Campinas (UNICAMP), Brazil.
>
> If you do not solely control copyright in the requested materials, I would
> appreciate any information you can provide about others to whom I should
> write.
>
> Best regards,
> Vagner Figueredo de Santana (santana.vagner@gmail.com), PhD Student
> M. Cecilia C. Baranauskas (cecilia@ic.unicamp.br), Professor at the
> Institute of Computing
>
> Institute of Computing - UNICAMP
> Albert Einstein Av., 1251, 13084-971
> Campinas - SP, Brazil
> Phone: +55 19 3521-5845
>

1 de 1 25/07/2012 02:14



ASSOCIATION FOR COMPUTING MACHINERY, INC. LICENSE TERMS AND CONDITIONS

Jul 26, 2012

This is a License Agreement between Vagner Santana ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Association for Computing Machinery, Inc., and the payment terms and conditions.

License Number	2956721348713
License date	Jul 26, 2012
Licensed content publisher	Association for Computing Machinery, Inc.
Licensed content publication	Proceedings of the International Cross-Disciplinary Conference on Web Accessibility
Licensed content title	Web accessibility and people with dyslexia: a survey on techniques and guidelines
Licensed content author	Vagner Figueredo de Santana, et al
Licensed content date	Apr 16, 2012
Type of Use	Thesis/Dissertation
Requestor type	Author of this ACM article
Is reuse in the author's own new work?	Yes
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Order reference number	
Title of your thesis/dissertation	Interfaces Autoajustáveis em Websites: Contribuições em Direção ao Design para Todos
Expected completion date	Oct 2012
Estimated size (pages)	150
Billing Type	Credit Card
Credit card info	Master Card ending in 6111
Credit card expiration	11/2014
Total	8.00 USD
Terms and Conditions	

Rightslink Terms and Conditions for ACM Material

1. The publisher of this copyrighted material is Association for Computing Machinery, Inc. (ACM). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at).

2. ACM reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

3. ACM hereby grants to licensee a non-exclusive license to use or republish this ACM-copyrighted material* in secondary works (especially for commercial distribution) with the stipulation that consent of the lead author has been obtained independently. Unless otherwise stipulated in a license, grants are for one-time use in a single edition of the work, only with a maximum distribution equal to the number that you identified in the licensing process. Any additional form of republication must be specified according to the terms included at the time of licensing.

*Please note that ACM cannot grant republication or distribution licenses for embedded third-party material. You must confirm the ownership of figures, drawings and artwork prior to use.

4. Any form of republication or redistribution must be used within 180 days from the date stated on the license and any electronic posting is limited to a period of six months unless an extended term is selected during the licensing process. Separate subsidiary and subsequent republication licenses must be purchased to redistribute copyrighted material on an extranet. These licenses may be exercised anywhere in the world.

5. Licensee may not alter or modify the material in any manner (except that you may use, within the scope of the license granted, one or more excerpts from the copyrighted material, provided that the process of excerpting does not alter the meaning of the material or in any way reflect negatively on the publisher or any writer of the material).

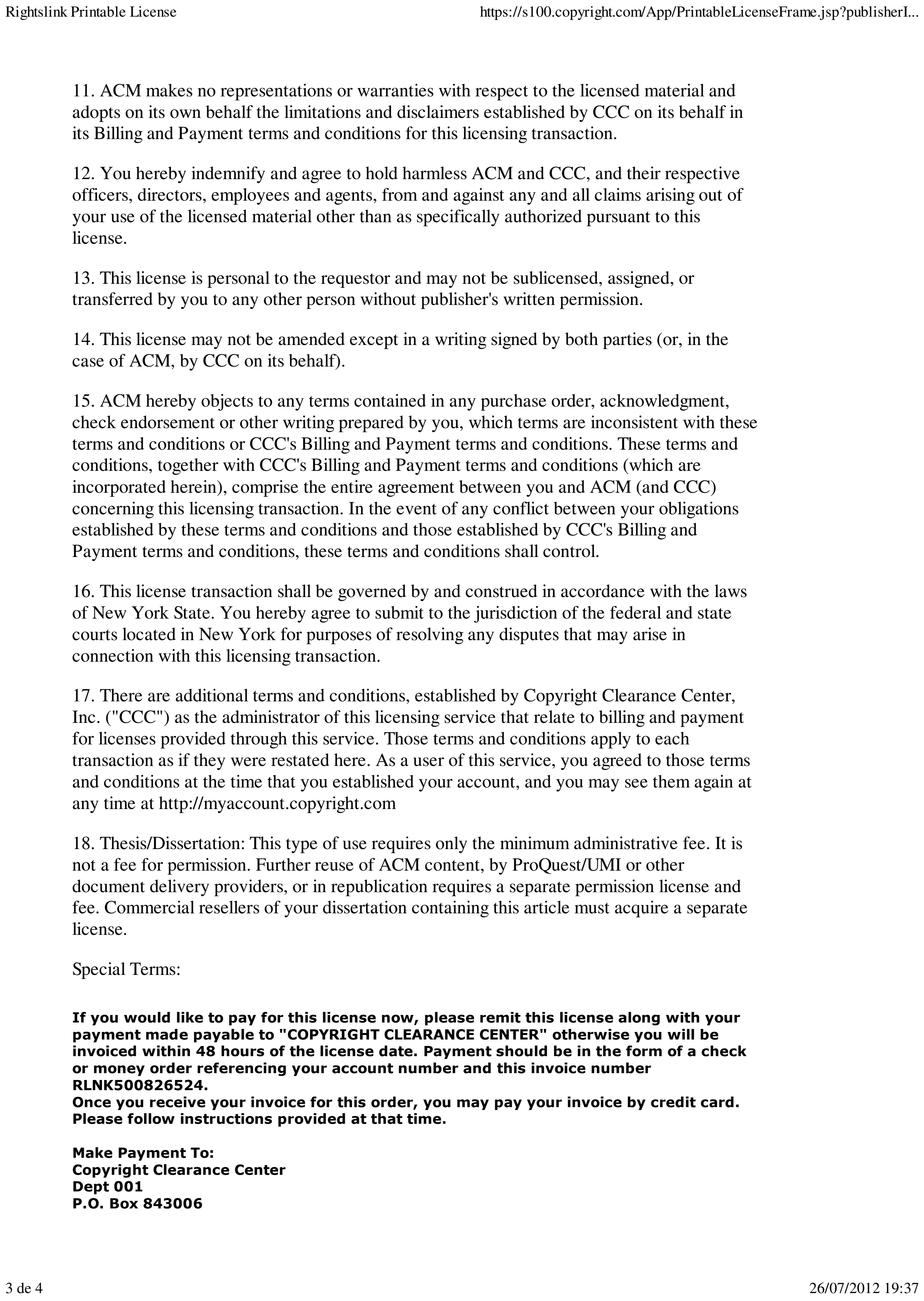
6. Licensee must include the following copyright and permission notice in connection with any reproduction of the licensed material: "[Citation] © YEAR Association for Computing Machinery, Inc. Reprinted by permission." Include the article DOI as a link to the definitive version in the ACM Digital Library. Example: Charles, L. "How to Improve Digital Rights Management," Communications of the ACM, Vol. 51:12, © 2008 ACM, Inc. http://doi.acm.org/10.1145/nnnnnn.nnnnnn (where nnnnnn.nnnnnn is replaced by the actual number).

7. Translation of the material in any language requires an explicit license identified during the licensing process. Due to the error-prone nature of language translations, Licensee must include the following copyright and permission notice and disclaimer in connection with any reproduction of the licensed material in translation: "This translation is a derivative of ACM-copyrighted material. ACM did not prepare this translation and does not guarantee that it is an accurate copy of the originally published work. The original intellectual property contained in this work remains the property of ACM."

8. You may exercise the rights licensed immediately upon issuance of the license at the end of the licensing transaction, provided that you have disclosed complete and accurate details of your proposed use. No license is finally effective unless and until full payment is received from you (either by CCC or ACM) as provided in CCC's Billing and Payment terms and conditions.

9. If full payment is not received within 90 days from the grant of license transaction, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted.

10. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.



11. ACM makes no representations or warranties with respect to the licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

12. You hereby indemnify and agree to hold harmless ACM and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

13. This license is personal to the requestor and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

14. This license may not be amended except in a writing signed by both parties (or, in the case of ACM, by CCC on its behalf).

15. ACM hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and ACM (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

16. This license transaction shall be governed by and construed in accordance with the laws of New York State. You hereby agree to submit to the jurisdiction of the federal and state courts located in New York for purposes of resolving any disputes that may arise in connection with this licensing transaction.

17. There are additional terms and conditions, established by Copyright Clearance Center, Inc. ("CCC") as the administrator of this licensing service that relate to billing and payment for licenses provided through this service. Those terms and conditions apply to each transaction as if they were restated here. As a user of this service, you agreed to those terms and conditions at the time that you established your account, and you may see them again at any time at http://myaccount.copyright.com

18. Thesis/Dissertation: This type of use requires only the minimum administrative fee. It is not a fee for permission. Further reuse of ACM content, by ProQuest/UMI or other document delivery providers, or in republication requires a separate permission license and fee. Commercial resellers of your dissertation containing this article must acquire a separate license.

Special Terms:

If you would like to pay for this license now, please remit this license along with your payment made payable to "COPYRIGHT CLEARANCE CENTER" otherwise you will be invoiced within 48 hours of the license date. Payment should be in the form of a check or money order referencing your account number and this invoice number RLNK500826524.
Once you receive your invoice for this order, you may pay your invoice by credit card. Please follow instructions provided at that time.

Make Payment To:
Copyright Clearance Center
Dept 001
P.O. Box 843006

Boston, MA 02284-3006

For suggestions or comments regarding this order, contact RightsLink Customer Support: customercare@copyright.com or +1-877-622-5543 (toll free in the US) or +1-978-646-2777.

Gratis licenses (referencing $0 in the Total field) are free. Please retain this printable license for your reference. No payment is required.

Bibliography

[1] Graphviz - Graph Visualization Software. http://www.graphviz.org.

[2] Section 508. http://www.section508.gov/.

[3] Julio Abascal and Luis Azevedo. Fundamentals of inclusive hci design. In *Proceedings of the 4th international conference on Universal access in human computer interaction: coping with diversity*, UAHCI'07, pages 3–9, Berlin, Heidelberg, 2007. Springer-Verlag.

[4] Julio Abascal and Colette Nicolle. Moving towards inclusive design guidelines for socially and ethically aware HCI. *Interacting with Computers*, 17(5):484–505, 2005.

[5] Areej Al-Wabil, Panayiotis Zaphiris, and Stephanie Wilson. Web navigation for individuals with dyslexia: an exploratory study. In *Proceedings of the 4th international conference on Universal access in human computer interaction: coping with diversity*, UAHCI'07, pages 593–602, Berlin, Heidelberg, 2007. Springer-Verlag.

[6] American Psychiatric Association APA. *Diagnostic and Statistical Manual of Mental Disorders.* Text Revision, 4th edition, 2000.

[7] Ernesto Arroyo, Ted Selker, and Willy Wei. Usability tool for analysis of web designs using mouse tracks. In *Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems*, volume 2 of *Work-in-progress*, pages 484–489, 2006.

[8] Richard Atterer and Albrecht Schmidt. Tracking the interaction of users with ajax applications for usability testing. In Mary Beth Rosson and David J. Gilmore, editors, *CHI*, pages 1347–1350. ACM, 2007.

[9] Sandrine Balbo, Steve Goschnick, Derek Tong, and Cécile Paris. Leading web usability evaluations to wauter. In *AusWeb05 - Australian World Wide Web Conference 2005*, 2005.

[10] M. C. C. Baranauskas, H. H. Hornung, and M. C. Martins. Design socialmente responsável: Desafios de interface de usuário no contexto brasileiro. In *Anais do XXVIII Congresso da Sociedade Brasileira de Computação*, pages 91–105, 2008.

[11] British Dyslexia Association BDA. Good practice guidelines for supporting employees with dyslexia in the workplace. `http://www.nafd.org.uk/media/150667/bda\%20employer\%20code_of_practice.pdf`, 2009.

[12] N. Beauchamp. Case study of a dyslexic person's visual perceptual problems: A fizz effect. `http://www.engsc.ac.uk/downloads/Disability/fizz.pdf`.

[13] Jeffrey P. Bigham, Anna Cavender, Jeremy T. Brudvik, Jacob O. Wobbrock, and Richard E. Ladner. Webinsitu: a comparative analysis of blind and sighted browsing behavior. In Enrico Pontelli and Shari Trewin, editors, *ASSETS*, pages 51–58. ACM, 2007.

[14] John Bradford. Dyslexia online magazine – designing web pages for dyslexic readers. `http://www.dyslexia-parent.com/mag35.html`, 2010.

[15] Giorgio Brajnik. Using automatic tools in accessibility and usability assurance processes. In *Lecture Notes in Computer Science Proceedings of the 8th ERCIM UI4ALL Workshop*. Springer Verlag, 2004.

[16] I. V. Cadez, D. Heckerman, P. Smyth, C. Meek, and S. White. Model-based clustering and visualization of navigation patterns on a Web site. *Data Mining and Knowledge Discovery*, 2003.

[17] Tonio Carta, Fabio Paternò, and Vagner Figuerêdo de Santana. Web usability probe: a tool for supporting remote usability evaluation of web sites. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part IV*, INTERACT'11, pages 349–357, Berlin, Heidelberg, 2011. Springer-Verlag.

[18] Manuel Castells. *A sociedade em rede – A era da informação: economia, sociedade e cultura*, volume 1. Editora Paz na Terra, 4 edition, 1999.

[19] José C. Castillo, H. Rex Hartson, and Deborah Hix. Remote usability evaluation: Can users report their own critical incidents? In *CHI Late-Breaking Results*, pages 253–254, 1998.

[20] Centro de Estudos sobre as Tecnologias da Informação e da Comunicação CETIC.BR. Pesquisa sobre o uso das tecnologias de informação e de comunicação no brasil. `http://www.cetic.br/usuarios/tic/2008/analise-tic-domicilios2008.pdf`, 2008.

[21] Ed Huai-hsin Chi, Adam Rosien, and Jeffrey Heer. Lumberjack: Intelligent discovery and analysis of web user traffic composition. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD*, volume 2703 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2002.

[22] Thiam Kian Chiew and Siti Salwa Salim. Webuse: Website usability evaluation tool. In *Malaysian Journal of Computer Science*, volume 16, pages 47–57, 2003.

[23] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, 2001. ACM.

[24] Bettye Rose Connell, Mike Jones, Ron Mace, Jim Mueller, Abir Mullick, Elaine Ostroff, Jon Sanford, Ed Steinfeld, Molly Story, and Gregg Vanderheiden. The principles of universal design. `http://www.ncsu.edu/project/design-projects/udi/center-for-universal-design/the-principles-of-universal-design/`, 1997.

[25] Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. Discovery of interesting usage patterns from Web data. *Lecture Notes in Computer Science*, 1836:163–182, 2000.

[26] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 1990.

[27] Francesco Correani, Barbara Leporini, and Fabio Paternò. Automatic inspection-based support for obtaining usable web sites for vision-impaired users. volume 5, pages 82–95. Springer-Verlag, 2006.

[28] Douglas Crockford. Jsonrequest. `http://json.org/JSONRequest.html`, 2006.

[29] Douglas Crockford. The <module> tag. `http://www.json.org/module.html`, 2006.

[30] Douglas Crockford. The world's most misunderstood programming language has become the world's most popular programming language. `http://javascript.crockford.com/popular.html`, 2008.

[31] Vagner Figueredo de Santana, Maria Cecília Calani Baranauskas, and Marco Aurélio Amaral Henriques. A framework for web 2.0 secure widgets. pages 69–76. IADIS, 2011.

[32] Anna Dickinson, Peter Gregor, and Alan F. Newell. Ongoing investigation of the ways in which some of the problems encountered by some dyslexics can be alleviated using computer techniques. In *Proceedings of the fifth international ACM conference on Assistive technologies*, Assets '02, pages 97–103, New York, NY, USA, 2002. ACM.

[33] R. Diestel. *Graduate Texts in Mathematics: Graph Theory*. Springer, 2nd edition, 2000.

[34] Dojo Toolkit. Cross domain XMLHttpRequest using an IFrame Proxy. `http://dojotoolkit.org/node/87`, 2006.

[35] José M. Domenech and Javier Lorenzo. A tool for web usage mining. In Hujun Yin, Peter Tiño, Emilio Corchado, William Byrne, and Xin Yao, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2007, 8th International Conference, Birmingham, UK, December 16-19, 2007, Proceedings*, volume 4881 of *Lecture Notes in Computer Science*, pages 695–704. Springer, 2007.

[36] Peter Eckersley. How unique is your browser? 2010.

[37] Michael Etgen and Judy Cantor. What does getting WET (web event-logging tool) mean for web usability? In *Proceedings of 5th Conference on Human Factors & the Web*, 1999.

[38] Kurt D. Fenstermacher and Mark Ginsburg. Mining client-side activity for personalization. In *WECWIS*, pages 205–212, 2002.

[39] John Clemans Flanagan. *The critical incident technique*. American Psychological Association, 1954.

[40] Martin Fowler. *Patterns of Enterprise Application Architecture*. Pearson Education, Inc., 1 edition, 2003.

[41] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object Oriented Software*. Addison Wesley, 1995.

[42] Joydeep Ghosh. *Scalable Clustering*. Lawrence Erlbaum Associates, 2003.

[43] Google. Google analytics. `http://www.google.com/analytics`, 2009.

[44] Monty Hammontree, Paul Weiler, and Nandini Nayak. Remote usability testing. *interactions*, 1(3):21–25, jul 1994.

[45] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[46] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. The MIT Press, 2001.

[47] S. M. Handler and W. M. Fierson. Learning disabilities, dyslexia, and vision. Technical Report 127, e818 Pediatrics, February 2011.

[48] H. Rex Hartson, Jose C. Castillo, John Kelso, and Wayne C. Neale. Remote evaluation: The network as an extension of the usability laboratory. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Evaluation*, pages 228–235, 1996.

[49] Birgit Hay, Geert Wets, and Koen Vanhoof. Mining navigation patterns using a sequence alignment method. *Knowledge and Information Systems*, 6(2):150–163, 2004.

[50] Elaine Cristina Saito Hayashi, Vânia Paula de Almeida Neris, Leonelo Dell Anhol Almeida, Carla Lopes Rodriguez, Maria Cecília Martins, and Maria Cec´ilia Calani Baranauskas. Inclusive social networks: Clarifying concepts and prospecting solutions for e-Cidadania. Technical Report IC-08-29, Institute of Computing, University of Campinas, October 2008.

[51] David M. Hilbert and David F. Redmiles. Extracting usability information from user interface events. *ACM Comput. Surv.*, 32(4):384–421, 2000.

[52] Jason I. Hong, Jeffrey Heer, Sarah Waterson, and James A. Landay. Webquilt: A proxy-based approach to remote web usability testing. *ACM Transactions on Information Systems*, 19(3):263–285, 2001.

[53] Instituto Brasileiro de Geografia e Estatística IBGE. Censo 2000. `http://www.ibge.gov.br/`, 2000.

[54] International Dyslexia Association IDA. What is dyslexia? `http://www.interdys.org/FAQWhatIs.htm`, 2007.

[55] International Dyslexia Association IDA. Dyslexia basics. `http://www.interdys.org/ewebeditpro5/upload/BasicsFactSheet.pdf`, 2008.

[56] Web Accessibility in Mind WebAIM. Survey of preferences of screen readers users. `http://webaim.org/projects/screenreadersurvey/`.

[57] Instituto Paulo Montenegro IPM. Boas notícias e um forte alerta são as principais revelações de inaf 2009. `http://www.ipm.org.br/ipmb_pagina.php?mpg=4.03.00.00.00&ver=por`, 2009.

[58] International Organization for Standardization ISO. Ergonomic requirements for office work with display terminals (VDTs). part 11: Guidance on usability, 1998.

[59] Melody Y. Ivory and Marti A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33(4):470–516, 2001.

[60] Collin Jackson and Helen J. Wang. Subspace: secure cross-domain communication for web mashups. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 611–620, New York, NY, USA, 2007. ACM.

[61] JGraphT, 2005.

[62] Kecheng Liu. *Semiotics in Information Systems Engineering.* Cambridge University Press, 2000.

[63] Víctor López-Jaquero, Francisco Montero, and Fernando Real. Designing user interface adaptation rules with t: Xml. In *Proceedings of the 14th international conference on Intelligent user interfaces*, IUI '09, pages 383–388, New York, NY, USA, 2009. ACM.

[64] John Maeda. *The Laws of Simplicity (Simplicity: Design, Technology, Business, Life).* MIT Press, 1 edition, 2006.

[65] Giorgio Maone. Add-ons for firefox – noscript. `https://addons.mozilla.org/pt-BR/firefox/addon/noscript/`, 2010.

[66] Model-Based UI XG MBUI-XG. Model-based ui xg final report. `http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui-20100504/`, 2010.

[67] Jacob E. McCarthy and Sarah J. Swierenga. What we know about dyslexia and web accessibility: a research review. *Univers. Access Inf. Soc.*, 9(2):147–152, jun 2010.

[68] Scott McCloud. *Understanding Comics: The Invisible Art.* William Morrow Paperbacks.

[69] Alexander Mikroyannidis and Babis Theodoulidis. A theoretical framework and an implementation architecture for self adaptive web sites. In *Web Intelligence*, pages 558–561. IEEE Computer Society, 2004.

[70] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, aug 2000.

[71] Anders Mϕrch. Three levels of end-user tailoring: Customization, integration, and extension. In *3rd Decennial Aarhus Conference*, Aarhus, Denmark, 1995.

[72] Petra Mutzel and Peter Eades. Graphs in software visualization - introduction. In Stephan Diehl, editor, *Software Visualization*, volume 2269 of *Lecture Notes in Computer Science*, pages 285–294. Springer, 2001.

[73] C. Narain. Childhood developmental disorders. *Nature Neuroscience*, 2006.

[74] Jakob Nielsen. Guerrilla hci: Using discount usability engineering to penetrate the intimidation barrier. `http://www.useit.com/papers/guerrilla_hci.html`, 1994.

[75] Jakob Nielsen. *Designing Web Usability: The Practice of Simplicity.* New Riders Publishing, Indianapolis, Indiana, USA, 2000.

[76] Jakob Nielsen. Customization of uis and products. `http://www.useit.com/alertbox/customization.html`, 2009.

[77] National Institute of Standards and Technology NIST. WebVIP. `http://zing.ncsl.nist.gov/WebTools/WebVIP/overview.html`, 2002.

[78] Adam A. Noureddine and Meledath Damodaran. Security in web 2.0 application development. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '08, pages 681–685, New York, NY, USA, 2008. ACM.

[79] Terri Oda, Glenn Wurster, P. C. van Oorschot, and Anil Somayaji. Soma: mutual approval for included content in web pages. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 89–98, New York, NY, USA, 2008. ACM.

[80] Open Icon Library OIL. Open icon library: free/open icons. `http://openiconlibrary.sourceforge.net`, 2010.

[81] Tim O'Reilly. What is web 2.0: Design patterns and business models for the next generation of software. `http://oreilly.com/web2/archive/what-is-web-20.html`, 2005.

[82] The Open Web Application Security Project OWASP. Owasp top ten project. `http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`, 2010.

[83] Laila Paganelli and Fabio Paternò. Intelligent analysis of user interactions with web applications. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 111–118, New York, NY, USA, 2002. ACM.

[84] Fabio Paternò, Angela Piruzza, and Carmen Santoro. Remote web usability evaluation exploiting multimodal information on user behavior. In Gaëlle Calvary, Costin Pribeanu, Giuseppe Santucci, and Jean Vanderdonckt, editors, *CADUI*, pages 287–298. Springer, 2006.

[85] Fabio Paternò and Carmen Santoro. Remote usability evaluation: Discussion of a general framework and experiences from research with a specific tool. In *Maturing Usability*, pages 197–221, London, 2008. Springer-Verlag.

[86] Charles Sanders Peirce. *Collected Papers of Charles Sanders Peirce.* Cambridge: Harvard University Press, 1974.

[87] Phu H. Phung, David Sands, and Andrey Chudnov. Lightweight self-protecting javascript. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 47–60, New York, NY, USA, 2009. ACM.

[88] Dimitrios Pierrakos, Georgios Paliouras, Christos Papatheodorou, and Constantine D. Spyropoulos. Web usage mining as a tool for personalization: A survey. *User Model. User-Adapt. Interact*, 13(4):311–372, 2003.

[89] Peter Rainger. A dyslexic perspective on e-content accessibility. `http://www.jisctechdis.ac.uk/techdis/resources/detail/learnersmatter/Jan03_Dyslexia`, 2003.

[90] Franck Ramus, Stuart Rosen, teven C. Dakin, Brian L. Day, Juan M. Castellote, Sarah White, and Uta Frith. Theories of developmental dyslexia: Insights from a multiple case study of dyslexic adults. (126):841–865, 2003.

[91] J. C. Reis, R. Bonacin, and M. C. C. Baranauskas. New perspectives for search in social networks: A challenge for inclusion. In *12th International Conference on Enterprise Information Systems (ICEIS)*, 2010.

[92] Heloisa Vieira da Rocha and Maria Cecilia Calani Baranauskas. *Design e Avaliação de Interfaces.* NIED/UNICAMP, 2003.

[93] Garett Rogers. Google analytics stops at 234,725 accounts. `http://www.zdnet.com/blog/google/google-analytics-stops-at-234725-accounts/36`, 2005.

[94] R. Romani and M. C. C. Baranauskas. Gwido - games with interaction design objective. In *IADIS-WWW/Internet 2009*, 2009.

[95] RSnake. Xss (cross site scripting) cheat sheetesp: for filter evasion. `http://ha.ckers.org/xss.html`, 2009.

[96] Jeffrey Rubin. *Handbook Of Usability Testing: How to plan, design, and conduct effective tests.* John Wiley & Sons Inc, 1st edition, 1994.

[97] Jesse Ruderman. The Same Origin Policy. `http://developer.mozilla.org/En/Same_origin_policy_for_JavaScript`, 2008.

[98] J. F. Sales, M. A. M. P. Parente, and S. S. Machado. As dislexias de desenvolvimento: Aspectos neuropsicológicos e cognitivos. *Interações*, 9(17):109–132, 2004.

[99] R. Santana. In *Métodos de Intervención en la Neuropsicología Infantil*, Puebla, México, 2001. Benemérita Universidad Autónoma de Puebla.

[100] Vagner Figuerêdo de Santana. Identificação de padrões de utilização da web mediada por tecnologias assistivas. Master's thesis, Instituto de Computação - Universidade Estadual de Campinas (UNICAMP), 2009. Supervisor: Profa. Maria Cecilia Calani Baranauskas.

[101] Vagner Figuerêdo de Santana, Leonelo Dell Anhol Almeida, and Maria Cecilia Calani Baranauskas. Aprendendo sobre acessibilidade e construção de websites para todos. *Revista Brasileira de Informática na Educação*, 16(3):71–83, 2008.

[102] Vagner Figuerêdo de Santana and Maria Cecilia Calani Baranauskas. An asynchronous client-side event logger model. Technical report, Institute of Computing (UNICAMP), 2008.

[103] Vagner Figuerêdo de Santana and Maria Cecilia Calani Baranauskas. A prospect of websites evaluation tools based on event logs. In Peter Forbrig, Fabio Paternò, and Annelise Mark Pejtersen, editors, *Human-Computer Interaction Symposium*, volume 272 of *IFIP International Federation for Information Processing*, pages 99–104. Springer Boston, 2008.

[104] Vagner Figuerêdo de Santana and Maria Cecilia Calani Baranauskas. Bringing users of a digital divide context to website evaluation using welfit. In *Fatores Humanos em Sistemas Computacionais (IHC 2010)*. SBC, 2010.

[105] Vagner Figuerêdo de Santana and Maria Cecilia Calani Baranauskas. Summarizing observational client-side data to reveal web usage patterns. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1219–1223, New York, NY, USA, 2010. ACM.

[106] Vagner Figuerêdo de Santana, Maria Cecilia Calani Baranauskas, and Marco Aurélio Amaral Henriques. Javascript na web 2.0: ameaças e prevenções no lado do cliente. Technical report, Institute of Computing (UNICAMP), 2010.

[107] Vagner Figueredo de Santana, Rosimeire de Oliveira, Leonelo Dell Anhol Almeida, and Maria Cecília Calani Baranauskas. Web accessibility and people with dyslexia: a survey on techniques and guidelines. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*, W4A '12, pages 35:1–35:9, New York, NY, USA, 2012. ACM.

[108] Vagner Figuerêdo de Santana, Diego Samir Melo-Solarte, Vânia Paula de Almeida Neris, Leonardo Cunha de Miranda, and Maria Cecilia Calani Baranauskas. Redes sociais online: Desafios e possibilidades para o contexto brasileiro. In *SEMISH - XXXVI Seminário Integrado de Software e Hardware*. SBC, 2008.

[109] B. B. Sawaia. *O Sofrimento Ético-Político Como Categoria de Análise da Dialética Exclusão/Inclusão*. Editora Vozes, 2004.

[110] John G. Schoeberlein and Yuanqiong (Kathy) Wang. Groupware accessibility for persons with disabilities. In *Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction. Part III: Applications and Services*, UAHCI '09, pages 404–413, Berlin, Heidelberg, 2009. Springer-Verlag.

[111] Cyrus Shahabi and Farnoush Banaei-Kashani. A framework for efficient and anonymous web usage mining based on client-side tracking. In *Lecture Notes in Artificial Intelligence: Proceedings of WEBKDD 2001*, volume 2356 of *Lecture Notes in Computer Science*, pages 113–144. Springer-Verlag, 2001.

[112] S.E. Shaywitz and B. A. Shaywitz. The neurobiology of reading and dyslexia. 5(A), 2001.

[113] M. Spiliopoulou and L. C. Faulstich. WUM: A tool for Web utilization analysis. *Lecture Notes in Computer Science*, 1590:184–203, 1999.

[114] Ronald Stamper, Kecheng Liu, Mark Hafkamp, and Yasser Ades. Understanding the roles of signs and norms in organizations - a semiotic approach to information systems design. *Behaviour Information Technology*, 19(1):15–27, 2000.

[115] Ronald K. Stamper. A semiotic theory of information and information systems / applied semiotics. In *Invited papers for the ICL/University of Newcastle Seminar on 'Information', September 6–10, 1993*, 1993.

[116] Ronald K. Stamper. *Signs of Work: Semiosis and Information Processing in Organisations*. Walter de Gruyter, 1996.

[117] Ronald K. Stamper. Extending semiotics for the study of organisations. In *Proceedings of Conference on Semiotics and the Information Sciences*, 1998.

[118] Robert J. Sternberg and Elena L. Grigorenko. *Crianças rotuladas: O que é necessário saber sobre as dificuldades de aprendizagem.* Porto Alegre: Artes Médicas, 2003.

[119] W3TECHS WEB TECHNOLOGY SURVEYS. Usage of traffic analysis tools for websites. `http://w3techs.com/technologies/overview/traffic_analysis/all`, 2011.

[120] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Person Education, 2006.

[121] Nelson Uto and Sandro Pereira Melo. Minicurso: Vulnerabilidades em aplicações web e mecanismos de proteção. In *Anais do SBSeg 2010*, 2009.

[122] Ariel Vargas, Harold Weffers, and Heloísa Vieira Rocha. A method for remote and semi-automatic usability evaluation of web-based applications through users behavior analysis. In *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*, MB '10, pages 19:1–19:5, New York, NY, USA, 2010. ACM.

[123] World Wide Web Consortium W3C. WCAG - Web Content Accessibility Guidelines 2.0. `http://www.w3.org/TR/2008/CR-WCAG20-20080430/`, 2008.

[124] W3Schools. HTML DOM Event Object. `http://www.w3schools.com/jsref/dom_obj_event.asp`, 2011.

[125] Web Accessibility Initiative WAI. Introduction do web accessibility. `http://www.w3.org/WAI/intro/accessibility.php`, 2005.

[126] Jacques Wainer. Métodos de pesquisa quantitativa e qualitativa para a ciência da computação. In *Jornadas de Atualização em Informática (JAI)*. SBC, 2007.

[127] World Federation of Neurology WFN. Report of research group on developmental dyslexia and world illiteracy. Technical Report 18, Bulletin of the Orton Society, 1968.

[128] Marco A. A. Winckler, Carla M. D. S. Freitas, and Jose Valdeni de Lima. Usability remote evaluation for www. In *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*, Interactive Posters, 2000.

[129] Gunnar Eyal Wolf-Iszaevich. Prácticas seguras de programación para sistemas web. *Ventana Informática*, (21):91–111.

[130] World Wide Web Consortium. HTML 5 differences from HTML 4. `http://dev.w3.org/html5/html4-differences/`, 2009.

[131] Chuan Yue and Haining Wang. Characterizing insecure javascript practices on the web. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 961–970, New York, NY, USA, 2009. ACM.

[132] V. Zarach. Ten guidelines for improving accessibility for people with dyslexia. `http://wiki.cetis.ac.uk/Ten_Guidelines_for_Improving_Accessibility_for_People_with_Dyslexia`, 2002.