使用客户端重构的个性化Web可访问性

Alejandra Garrido,Sergio Firmenich,Gustavo Rossi,and Juliań Grigera
阿根廷 拉普拉塔国立大学
Nuria Medina-Medina
西班牙 格拉纳达大学
Ivana Harari
阿根廷 拉普拉塔国立大学

Abstract

According to W3C accessibility standards, most Web applications are neither accessible nor usable for people with disabilities. Developers often solve this problem by building separate accessible applications, but these are seldom usable and typically offer less functionality than the original. Another common solution is to maintain a single application, but create an accessible view by applying on-the-fl transformations to each requested page — a solution that rarely suits all audiences. A third solution is described here: let users improve Web accessibility in their client browsers through interface refactorings, which offer many customized, accessible views of a single application.通过W3C可访问性标准，大部分Web应用程序对于残疾人群都是不可访问且不可用的。开发者经常通过创建独立的可访问性应用程序来解决这个问题，但是那很少可用，并且通常提供的功能比原来少。另一个常用的解决方法是保持单一的应用程序，但是通过对每个请求的页面采用FL变换来创建一个可访问的视图，是一个难以适用于所有用户的解决方案。这里描述的是第三种解决方案：让用户通过界面重构提高他们客户端浏览器的Web可访问性，它提供了许多单个应用程序定制的可访问视图。

I. Introduction

Refactoring was originally conceived as a technique to improvesoftware’s internal qualities —-such as understandability and maintainability — while preserving semantics.1 In prior work, we adapted the refactoring approach to improve a Web application’s external attributes, such as usability.2 These Web refactorings consist of small navigation or interface transformations that enhance perceivable aspects of Web applications, such as user interaction and content presentation, while preserving functionality. Refactorings can also solve accessibility and usability problems for disabled users.3 Still, it’s usually impractical to address interface improvements for all audiences because disabilities can vary dramatically in nature (visual, cognitive, or motor), severity (blindness, color blindness, or strabismus) and extent (total or partial). In such different contexts, “one for all” is barely feasible.

When applying refactoring to improve internal qualities, developers decide which transformations to apply and where, because they’re the ones benefiting from the improvement. As Brian Foote and Joseph Yoder put it, “Who better to resolve the forces impinging upon each design issue as it arises, as the person who is going to have to live with these decisions?”4 Moreover, different developers might prefer alternative solutions for the same “bad smell” (that is, the design problem that motivated the refactoring1). Following on this general philosophy, we believe that end users should be able to tailor a website’s interface for their own benefit.

II. Methodologies

This document applies to version 1.7 and later of IEEEtran. Prior versions do not have all of the features described here.IEEEtran will display the version number on the user’s console when a document using it is being compiled. The latest version of IEEEtran and its support files can be obtained from IEEE’s web site, or CTAN . This latter site may have some additional material, such as beta test versions and files related to non-IEEE uses of IEEEtran. See the IEEEtran homepage for frequently asked questions and recent news about IEEEtran. Complimentary to this document. This document applies to version 1.7 and later of IEEEtran. Prior versions do not have all of the features described here.IEEEtran will display the version number on the user’s console when a document using it is being compiled. The latest version of IEEEtran and its support files can be obtained from IEEE’s web site, or CTAN . This latter site may have some additional material, such as beta test versions and files related to non-IEEE uses of IEEEtran. See the IEEEtran homepage for frequently asked questions and recent news about IEEEtran. Complimentary to this document. This document applies to version 1.7 and later of IEEEtran. Prior versions do not have all of the features described here.IEEEtran will display the version number on the user’s console when a document using it is being compiled. The latest version of IEEEtran and its support files can be obtained from IEEE’s web site, or CTAN . This latter site may have some additional material, such as beta test versions and files related to non-IEEE uses of IEEEtran. See the IEEEtran homepage for frequently asked questions and recent news about IEEEtran. Complimentary to this document. This document applies to version 1.7 and later of IEEEtran. Prior versions do not have all of the features described here.IEEEtran will display the version number on the user’s console when a document using it is being compiled. The latest version of IEEEtran and its support files can be obtained from IEEE’s web site, or CTAN . This latter site may have some additional material, such as beta test versions and files related to non-IEEE uses of IEEEtran. See the IEEEtran homepage for frequently asked

questions and recent news about IEEEtran. Complimentary to this document.