



*Centrale
Nantes*

Web Strategy Project

Accessibilité de la plateforme eZoomBook
pour les malvoyants

**Félix Tournoux
Ibrahima Ouattara
24/04/2014**



Project Outline

Organisation's name and address: eZoomBook.com

Email: mayleen.richard-lacouture@ec-nantes.fr

Field of activity: exchange and collaboration around reading.

Name of the supervisor within the partner organisation: Mrs. Mayleen Richard-Lacouture

Name of students: Felix TOURNOUX, Ibrahima OUATTARA

Objective (s): Make the EZoombook website accessible to blind people.

EZoomBook is a platform for exchange and collaboration around reading, offering innovative digital book format.

So are involved in project work in two parts , theoretical and practical :

-Theoretical issue : state of the art technologies for visually impaired , analysis recommendations for accessibility, examples of non Matching / sites tips for making a site, etc. . This file must be a kit to educate and train programmers to issues of accessibility for the visually impaired .

-Practical work: test eZoomBook platform, make a diagnosis in collaboration with the Association for the Blind avh (www.avh.asso.fr)

Project Articulation:

Make the EZoombook website accessible to blind people.

Technical Characteristics:

- Using an appropriate software (total validator, colorcontrast analyser,..) to analyse the website.
- Add some functionalities like function to search books on the platform.
- Making the platform more interactive
- Internalization of the website

- Adding language markers
- *Developing the site*
- Developing appropriate web services with respect to interoperability, accessibility, ergonomics
- Applying a quality assurance method (testing and afferent procedures)
- Complying with the IT legal issues ...

Schedule and intermediary objectives (for the 2 meetings with your tutor):

Objetive of the 1st meeting:

We briefly described the aim of the project and how we think we will carry it.

Advices of our web-development tutor:

- establish a good schedule
- Try to have an idea about previous study of ezoombook website.
- Try to run play on our localhost.
- get the code source of ezoombook website and work with it
- learn Scala : The ezoombook website is coded under Scala that we don't master. Given we are following courses in Java, we thought it would be relevant to follow Scala to elaborate our codes.
- send to our tutor an email to have the new version of the code.

Objetive of the 2nd meeting:

We explained the following different technical issues we had encountered.

- We are eventually 2 to work on the project.
- Mayleen had not answered to our emails (in fact we emailed her before to have some idea about the database acces in order to work on some functions but she had not answered)
- We decide to work on the Internalization of the websiteb which dose not require the database access. Just to avoid losing time while waiting for mayleen response.
- Take into account the recomendation of C. Evain : Make the plateforme more interactive by displaying some message while someone is uploading a book. Message like « please wait until your document has been saved before you click on other page »

Etat de l'art des technologies d'assistances

Aujourd'hui il existe plusieurs technologies d'assistances aux personnes aveugles et malvoyantes.

Lecteurs d'écran et navigateurs vocaux

En donnant des informations via une voie synthétique et le braille, ces lecteurs permettent aux personnes aveugles ou malvoyantes d'accéder à un ordinateur sans coût additionnel par rapport à une personne voyante.

Les lecteurs les plus utilisés sont :

JAWS (Job Access With Speech) créé par Freedom Scientific

NVDA (NonVisual desktop Access), qui est une revue d'écran gratuite et open-source pour le système d'exploitation Microsoft windows. Il est développé par NV Access, avec contribution de l'ensemble des utilisateurs.

VoiceOver

Cette technologie va au-delà de la simple description de ce qu'il se passe. Elle permet de faire bouger les choses. Elle guide aussi l'utilisateur à effectuer des tâches, comme sélectionner une option de menu ou activer un bouton à l'aide du clavier ou trackpad. VoiceOver confère un contrôle absolu de Mac (geste, fonction virtuelle rotor,..)

VoiceOver est sans équivalent, car ce n'est pas un lecteur d'écran autonome. Il est intégré au cœur d'OS X. Par conséquent, dès que les développeurs actualisent leurs apps de façon à exploiter les interfaces d'accessibilité fournies par Apple, ces apps peuvent être utilisées avec VoiceOver.

Autres outils

Window-eye, Orca (lecteur d'écran et loupe), WebAnywhere (lecteur d'écran en ligne), Browsealoud, Firevox (extension Firefox), ZoomTest (agrandisseur et lecteur d'écran), Assistive technology Experiment , Dragon NaturallySpeaking

Aides à la navigation

Readbale Bookmartlet, Readability (extension Firefox),Google-axe-chrome (Chrome extension for accessibility)

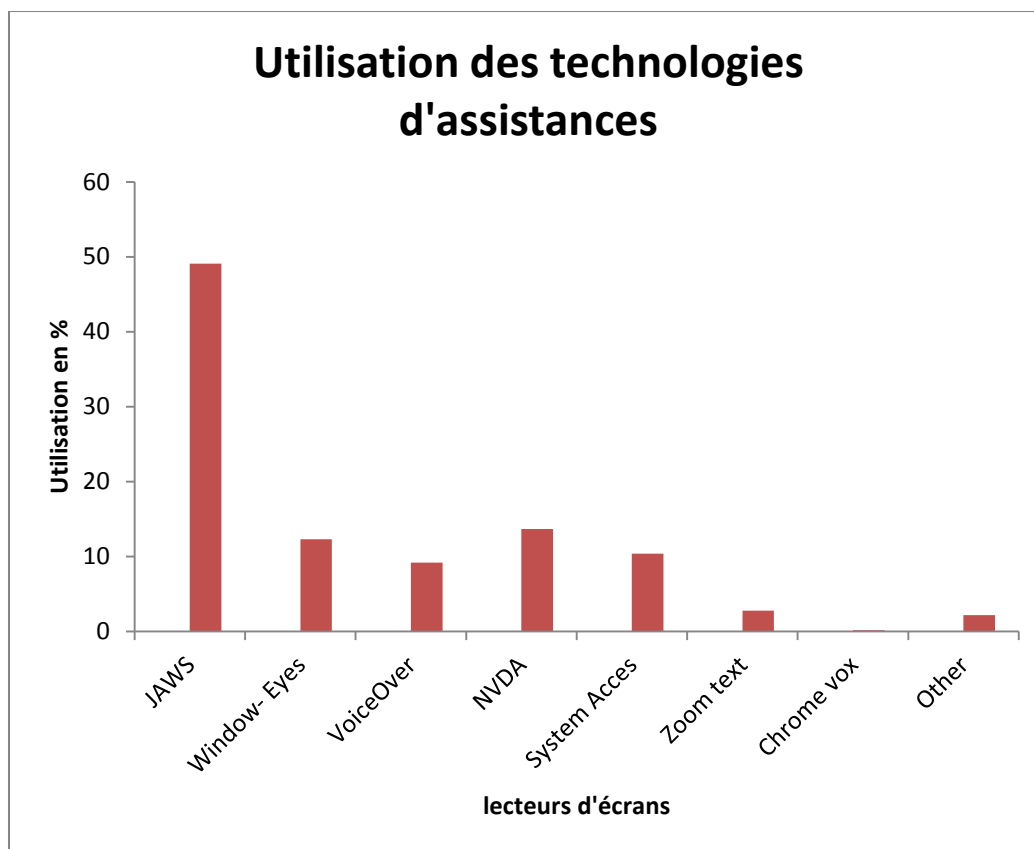
Services de personnalisation

Ces logiciels permettent à la fois, un accès direct optimisé pour un afficheur braille ou vocal, mais également une lisibilité améliorée et personnalisable des pages selon plusieurs paramètres: choix de la couleur du fond d'écran, taille et couleur des caractères, impression

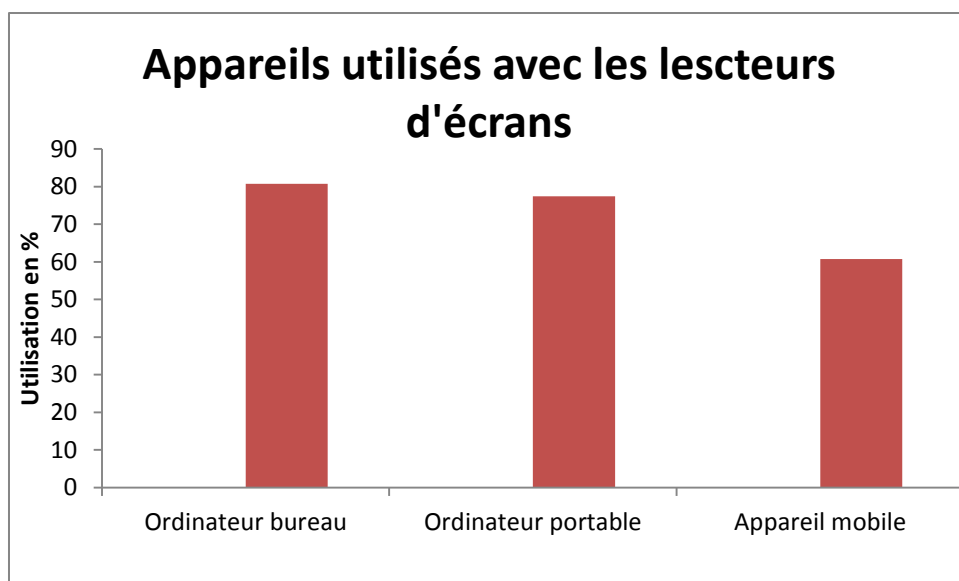
des pages conservant le profil défini. Les plus utilisés sont **Confort de lecture** (est open-source) et **Handilog**

Grâce à ces technologies toutes les personnes en état d'handicap peuvent avoir accès à l'information.

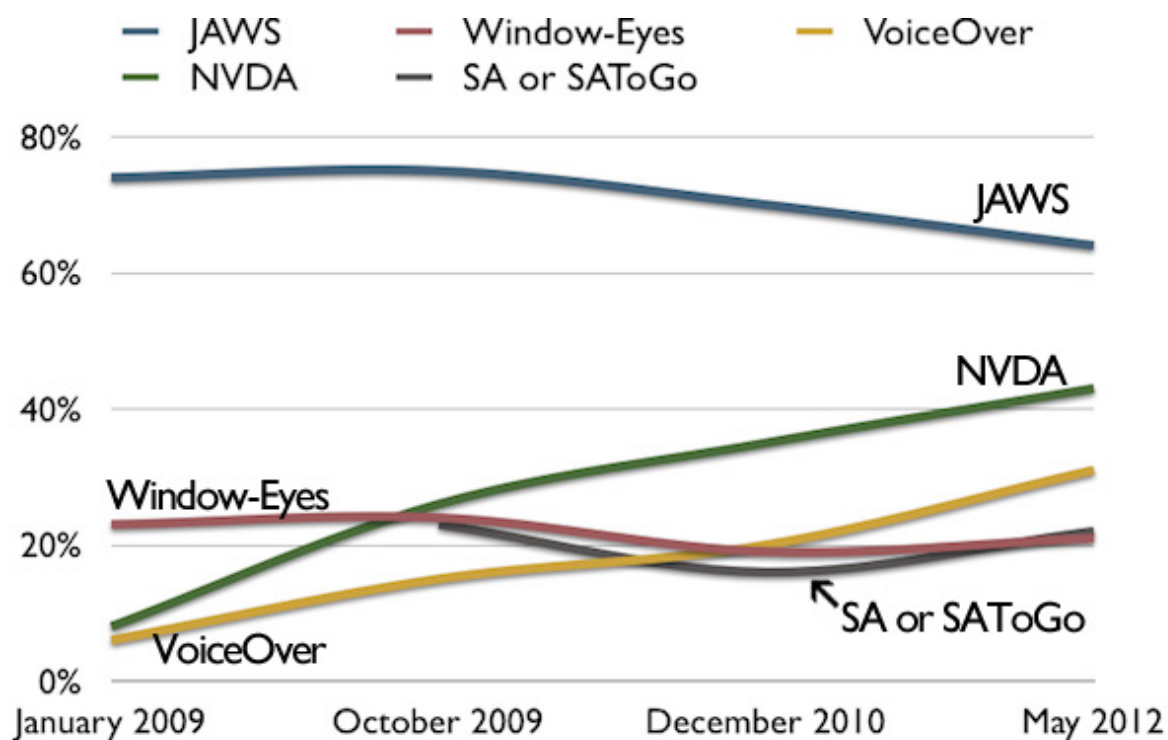
D'après une enquête WebAIM (Web Accessibility in Mind) en 2012. Nous avons plusieurs informations sur l'utilisation de ces technologies.



Toujours d'après cette enquête on peut voir que ces logiciels s'intègrent bien dans tous les appareils permettant d'accéder aux informations.



Evolution de l'utilisation de ces logiciels



Bien que JAWS soit le plus utilisé comme lecteur d'écran, on peut voir que le nombre d'utilisateurs est passé de 72% en 2009 à 62% en Décembre 2012.

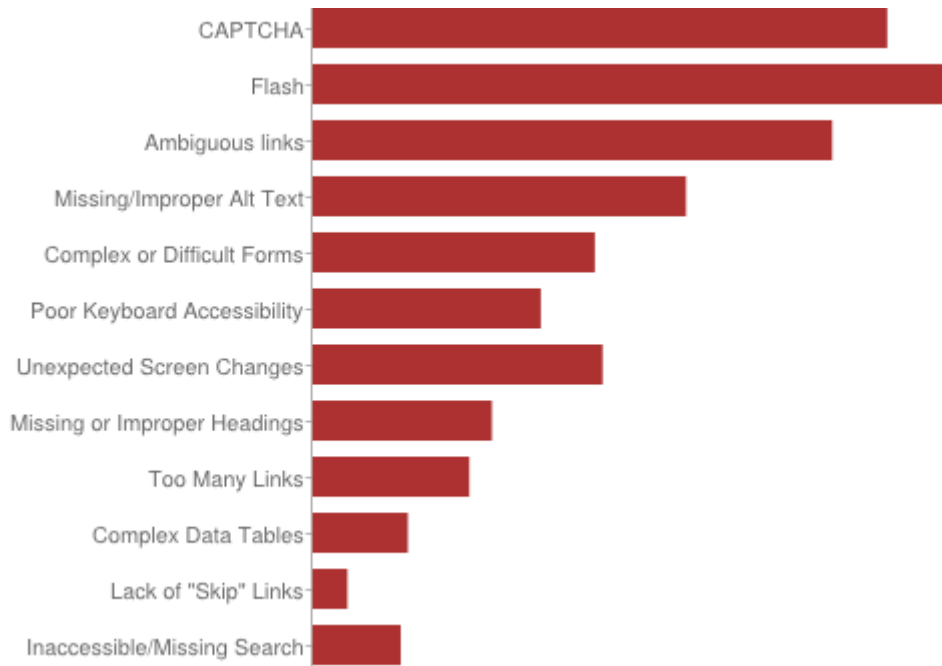
Contrairement à NVDA dont l'utilisation ne cesse de croître. Son utilisation passe de 8% en 2009 à 43% en 2012.

Cette tendance d'utilisation croissante s'observe aussi pour les autres logiciels utilisés après JAWS et NVDA.

En effet de plus en plus les utilisateurs abandonnent JAWS qui coûte cher au profit des autres technologies gratuites ou à prix très bas.

Même équipés de toutes ces technologies de pointe, les utilisateurs sont souvent en difficulté voire même frustrés face aux technologies de programmation. Ainsi les programmeurs doivent savoir et prendre en compte ces difficultés afin de leur faciliter l'accessibilité du web.

Ce diagramme montre les problématiques qui posent les plus de problèmes aux utilisateurs des technologies d'assistances.



Idée : dossier théorique : état de l'art des technologies pour mal voyant, analyse des recommandations pour l'accessibilité, exemples de sites compatibles/non compatibles, conseils pour la réalisation d'un site, etc. Ce dossier devra donc être un kit permettant de sensibiliser et de former des programmeurs aux problématiques d'accessibilité pour les mal voyants. Vous pourrez vous renseigner sur ce sujet grâce aux nombreuses sources que l'on peut trouver sur internet, mais aussi et surtout en contactant et en rencontrant monsieur Fernando Pinto (qui viendra prochainement faire une conférence à l'ECN).

Diagnostic du site existant

Nous avons analysé le site avec le logiciel Total Validator.

Ce logiciel nous a permis de détecter les principales faiblesses du site.

L'analyse de ces faiblesses est surtout basée sur l'accessibilité via des technologies d'assistance (terminal vocal, braille...)

On peut classer ces faiblesses suivant les trois principes du WCAG (Perception, utilisation et robustesse).

Présentation

- ✓ Toutes les images possèdent des alternatives textuelles.
- ✓ La présentation du contenu est linéaire : Il est impossible de contourner la liste des livres (topbooks) en utilisant la touche Tab, ce qui est embêtant.
 - Cependant si la liste des titres est à un niveau de titre supérieur au grand titre « Top Books », les outils à disposition des non-voyants permettent de sauter ces éléments.
- Trop de CSS et de javascript en ligne.

```
<style>
```

```
.rightpanel{  
  position: absolute;  
  top: 170px;  
  left: 60%;  
  right:15%  
}
```

```
.middlepanel{  
  position: absolute;  
  top: 170px;  
  right: 60%;  
  left: 35%;  
}
```

```
.leftpanel{  
  position: absolute;  
  top: 170px;  
  right: 65%;  
  left: 10%;  
}  
</style>
```

- L'information et la structure de présentation ne sont pas toujours séparées : le CSS est inclus dans du javascript ou dans le code html.
 - Ce qui ne facilite pas la navigation par les technologies d'assistance.
 - Comme solution on peut mettre tous les CSS et javascript respectivement dans des fichiers d'extension.css et .js pour ne dépendre que des attributs *id* et *class*.
- De même dans certain labels, les attributs *for* et *id* n'ont pas la même valeur.

- Dans certain ancres, les attributs *id* et *name* n'ont la même valeur
- La hiérarchie des balises Hn n'est pas toujours respectée. (saut de balise, commence pas toujours par h1, ...)

```

<h4>@Messages("parameters.modifylanguage.header")</h4>

    @helper.form(routes.Workspace.changelang){
      @select(langForm("locale"), options = Seq("fr" -> "French", "en" ->
"English"), '_label -> "")
      <div>
        <input type="submit"
value='@Messages("parameters.modifylanguage.submit")'>
      </div>
    }

    <a><h5>@Messages("parameters.unsubscribenewsletter.link")</h5></a>

    @defining(MD5Util.md5Hex(context.user.map(u =>
u.email).getOrElse("ezoomwiki.com"))){icon =>
      
    } <a
href="http://en.gravatar.com/emails/">@Messages("parameters.modifyimage.link")<a>

</div>
<div class="col-md-6">
  @if(error != ""){

    <div class="alert alert-info">
      <button type="button" class="close" data-dismiss="alert">x</button>
      @error
    </div>
  }else{}

<h4>@Messages("parameters.modifyinfo.header")</h4>

```

Nous avons ainsi dans un deuxième temps essayé de rétablir les balisages

```

diff --git a/app/views/bookedit.scala.html b/app/views/bookedit.scala.html
index be8e8fd..6ea6b3c 100644
--- a/app/views/bookedit.scala.html
+++ b/app/views/bookedit.scala.html
@@ -106,7 +106,7 @@ a.addlink<
<div class="row">
  <div class="col-md-12">
    <h2>Book metadata edition</h2>
    <h1>Book metadata edition</h1>
    @if(bookForm.hasErrors){
      <div class="alert alert-error">
        <a class="close" data-dismiss="alert">x</a>
diff --git a/app/views/bookreedit.scala.html b/app/views/bookreedit.scala.html
index 3a03610..de162bc 100644
--- a/app/views/bookreedit.scala.html
+++ b/app/views/bookreedit.scala.html
@@ -76,11 +76,11 @@ function toggle<divtag><
    @helper.form(routes.EzoomBooks.saveEditedBook){
      <h4>@inputText(bookForm("title"),'_label->"title")</h4>
      <h2>@inputText(bookForm("title"),'_label->"title")</h2>
      @hidden(bookForm("book_id"))
      @helper.repeat(bookForm("authors")){f =>
        <div id="authors@" style="margin-bottom:4px;" class="authors">
          <h4>@inputText(f,'_label->"author")</h4>
          <h2>@inputText(f,'_label->"author")</h2>
        </div>
      }
      <input type="button" id="btnAddAuth" value="Add another author" >
@@ -89,30 +89,30 @@ function toggle<divtag><
    <fieldset class="midlepanel">
      @helper.repeat(bookForm("languages")){f =>
        <div id="languages@" class="languages">
          <h4>@inputText(f,'_label->"languages")</h4>
          <h2>@inputText(f,'_label->"languages")</h2>
        </div>
      }
    }
  }
}

```

Utilisation

- ✓ Le titrage des pages est assez correct.
- ✓ La navigation au clavier fonctionne bien aussi.
- La fonction recherche de livre ne marche pas. Comme le montre ce bout du code.

```

<h1>Advanced search</h1>
</br> </br>
<fieldset>
  <center><form id="searchbox" action="" style="width:50%">
    <input id="search" type="text" placeholder="Search" style="width:75%">
    <input id="submit" type="submit" value="Search" style="width:15%">

  <form id="searchbox" action="" class="type">
    <input id="search" type="text" placeholder="Type">
    <input id="submit" type="submit" value="Search">

  </form>

```

En effet on remarque qu'il y a aucune action dédiée à cet effet.

- Nous chercherons une action à insérer dans ce form
- De même l'aide pour le mot de passe oublié ne fonctionne pas.

→

La page courante n'est pas marquée dans le menu de navigation. L'utilisateur n'est pas aidé à se repérer sur le site

eZoomBook



→ Nous pouvons modifier les CSS pour marquer les pages.

- Pass assez interactive. En effet dans la version actuelle, l'utilisateur n'est pas alerté pendant le téléchargement de son livre. Et si celui-ci change de page pendant le téléchargement, il reçoit un message d'erreur sans d'autres explications.
→ on peut afficher des petits messages à l'utilisateur pendant toute la durée de son téléchargement.

Robustesse

On note une absence des balises de langue comme le montre l'en-tête du code de la page principale ci-dessous.

```
<!DOCTYPE html>

<html>
  <head>
    <title>@title</title>
    <link rel="stylesheet" media="screen"
href="@routes.Assets.at("stylesheets/main.css")">
    <link rel="shortcut icon" type="image/png"
href="@routes.Assets.at("images/favicon.png")">
    <!-- Bootstrap -->
    <link rel="stylesheet" media="screen"
href="@routes.Assets.at("stylesheets/bootstrap/bootstrap.css")" />
    <!-- Additional styles -->
    @styles
    <!-- Masha for highlighting -->
    <link rel="stylesheet" type="text/css" href='@routes.Assets.at("stylesheets/masha.css")'
/>
  </head>
```

→ Plutôt que de mettre des balises de langues nous allons traduire les pages : internationaliser la plateforme.

Pour permettre à l'utilisateur de pouvoir choisir sa langue (Français, Anglais,...)

Internationalisation

L'internationalisation du site s'effectue en plusieurs étapes :

- Conversion de tout texte contenu dans les views en « message »
- Ces messages sont stockés dans conf/messages, avec le texte qui correspond en anglais
- Constitution d'un lexique dans conf/messages.fr avec la même structure et les traductions des messages en français
- La sélection de la langue se fait automatiquement par détection de l'origine de la connexion de l'utilisateur.
- Un menu déroulant sur le site permet de sélectionner une autre langue.

Maintenant que ce système est en place, il sera très facile de le proposer dans d'autres langues, car il suffit de créer le lexique, et d'ajouter la langue dans le fichier conf/application.conf

Lorsque l'on modifie le site et que l'on veut ajouter du texte, il ne faut pas le mettre directement dans le html, mais créer un nouveau message que l'on ajoute à la liste du fichier conf/messages

Lorsqu'un message n'existe pas dans une langue, il sera donné dans la langue par défaut, en anglais.

Remarque : Les caractères spéciaux ont une écriture particulière en UTF-8 :

http://www.e-bancel.com/codes_utf8_caracteres_accentues.php

Le problème doit pouvoir être évité avec certains éditeurs de texte, en tout cas il faut modifier le messages.fr

Rapport sur la situation fin avril 2014 :

Maxime Dô a créé une branche internationalisation dans laquelle il corrige des erreurs et rajoute des messages. A priori il ne travaille plus dessus. Il faut vérifier que tout fonctionne bien, et compléter le fichier conf/message.fr à partir de celui que nous avons fourni (j'en mets le contenu sur le blog) et du fichier conf/messages en anglais.

Amélioration

Après amélioration, tester la plateforme eZoomBook en collaboration avec l'association pour aveugles avh (voir contact ci-dessous et le site de l'association: www.avh.asso.fr/)

Biographie :

http://www.webaccessibilite.fr/liens-utiles-accessibilite-technologies-d-assistance.php#liens_1

<http://www.nvda-fr.org/>

<http://ezoombookresearch.com/blog/>

<http://www.playframework.com/>

<http://www.w3.org/standards/webdesign/accessibility>