Multilingual accessibility in human-screen reader interaction with web content: an exploratory study[1]

Álvaro García-Garcinuño
Jesús Torres-del-Rey

Álvaro García-Garcinuño
Universidad de Salamanca/Universidad de Puerto Rico;
alvaro.garcia1@upr.edu;
ORCID: 0000-0002-6070-1154

Jesús Torres-del-Rey
INICO, Universidad de Salamanca;
jtorres@usal.es;
ORCID: 0000-0001-6661-859X

Abstract

Blind people — among others with disabilities — use screen readers to interact with websites. This article examines how these assistive technologies deal with multilingual content, which is crucial for ensuring equitable access to multilingual information and activities for all. Drawing on García's (2020) pioneering study, which systematically analysed a wide range of screen readers in a multilingual web environment, we introduce a broader, more academic perspective, using qualitative analysis tools such as thematic coding. We have enhanced García's (2020) test pages, methodology, and analytical focus to identify specific translation-oriented challenges in the multilingual performance of screen readers and to classify them into a set of categories and subcategories of translation and localisation phenomena that may be useful in future research studies and interaction designs.

Keywords: multilingualism, web accessibility, screen readers, text-to-speech, web browsers, translation, localisation, human-computer interaction, thematic coding.

Resumen

Las personas ciegas —y otras personas con discapacidades— utilizan lectores de pantalla para interactuar con sitios web. Este artículo examina cómo estas tecnologías de apoyo gestionan el contenido multilingüe, crucial para garantizar un acceso equitativo a la información y a las actividades multilingües para todo el mundo. Basándonos en el estudio pionero de García (2020), que analizaba de manera sistemática una amplia gama de lectores de pantalla en un entorno web multilingüe, ofrecemos una perspectiva más amplia y académica mediante herramientas de análisis cualitativo como la codificación temática. Hemos ampliado las páginas de prueba, la metodología y el enfoque analítico de García (2020) para identificar retos específicos relacionados con la traducción en el rendimiento multilingüe de los lectores de pantalla y para clasificarlos en un conjunto de categorías y subcategorías de fenómenos de traducción y localización que pueden ser útiles en futuros estudios de investigación y en el diseño de interacciones.

Palabras clave: multilingüismo, accesibilidad web, lectores de pantalla, síntesis de texto a voz,

Rebuda: 7 de novembre de 2023 / Acceptació: 4 de novembre de 2024 / Publicació avançada: 20 de desembre de 2024

Citació recomanada: GARCÍA-GARCINUÑO, ÁLVARO; JESÚS TORRES-DEL-REY (2024). «Multilingual accessibility in human-screen reader interaction with web content: an exploratory study». Revista Tradumàtica. Tecnologies de la Traducció, 22, 426-449. <https://doi.org/10.5565/rev/tradumatica.380>

http://revistes.uab.cat/tradumatica
ISSN 1578-7559

navegadores web, traducción, localización, interacción persona-ordenador, codificación temática.

Resum

Les persones cegues —juntament amb altres persones amb discapacitats— utilitzen lectors de pantalla per interactuar amb llocs web. Aquest article examina com aquestes tecnologies assistives gestionen el contingut multilingüe, crucial per garantir un accés equitatiu a la informació i a les activitats multilingües per a tothom. Basant-nos en l'estudi pioner de García (2020), que analitzava de manera sistemàtica una àmplia gamma de lectors de pantalla en un entorn web multilingüe, oferim una perspectiva més àmplia i acadèmica mitjançant eines d'anàlisi qualitativa com la codificació temàtica. Hem millorat les pàgines de prova, la metodologia i el focus analític de García (2020) per identificar reptes específics relacionats amb la traducció en el rendiment multilingüe dels lectors de pantalla i per classificar-los en un conjunt de categories i subcategories de fenòmens de traducció i localització que poden ser útils en futurs estudis de recerca i en el disseny d'interaccions..

Paraules clau: multilingüisme, accessibilitat web, lectors de pantalla, conversió de text a veu, navegadors web, traducció, localització, interacció persona-ordinador, codificació temàtica .

1. Introduction

In June 2020, Xurxe Toivo García, an accessibility specialist, performed a series of multilingual experiments with screen readers (SRs) and published the article *The troubled state of screen readers in multilingual situations*, which immediately resonated with us as localisation and accessibility researchers, particularly when he lamented:

> as far as I know, this type of knowledge wasn't recorded anywhere. Firstly, it's already hard enough to find tutorials and instructions about how to set up screen readers so they even work in multilingual situations. And then there's the matter that's more relevant to my work: I couldn't find a simple source that discussed multilanguage bugs in screen readers, which is ridiculous considering that every single screen-reading software has quite a few of these (not to mention that screen reader users are as capable of speaking multiple languages as anyone else). (García, 2020)

García's data showed that language changes in speech synthesis are sometimes erratic and do not always follow the HTML `lang` attribute, according to the HTML standard (WHATWG, 2024b). This prevents equitable access to multilingual information and activities for all.

[1] This article is part of the PReLemma project (Parameters for more accessible multilingual lexical resources), funded by the Spanish Ministry of Science and Innovation (Project ID: PID2022-137210OB-I00).

1.1 Screen readers and multilingualism in the literature

In addition, García (2020) complained that there were no studies on the (poor) multilingual performance of SRs, a concern that became apparent during our research. To verify this, we conducted a thorough search using various keyword combinations, including "multilingual", "plurilingual", "screen reader", and "assessment", across multiple platforms: we used the Google search engine to explore professional blogs and articles, and Google Scholar, Scopus, and Clarivate's Web of Science[2] to search for relevant academic studies.

In the academic field, there were other articles that mentioned aspects related to our research but with different approaches and objectives. For example, Rodríguez Vázquez (2015) and Casalegno (2018) conducted user studies to identify the challenges faced by visually impaired individuals accessing multilingual content with SRs. Another study by Raghavendra & Prahallad (2010) drew attention to the technical limitations of building multilingual SRs in Indian languages. Finally, Ong (2022) explored the development of a multilingual text-to-speech synthesiser that would be able to quickly adapt new languages for a SR.

In the broader Google search, we found several references (blogs, support articles, forum discussions, etc.), mostly dealing with fairly general aspects of accessibility in multilingual websites, written by and for website developers — see, for example, Roselli (2023) or TPGi (2017).

In sum, our bibliographical review revealed the remarkable lack of literature on this specific topic. Thus, García (2020) was the only article that provided a systematic analysis of the performance of a wide range of browser-SR combinations in a multilingual web environment.

1.2 Screen readers, language and human-computer interaction

A SR is the assistive technology that most blind people — as well as others with visual and cognitive disabilities — use to access digital information. It reads the content displayed on the screen of a mobile phone, tablet, laptop or desktop computer, and converts it into speech or braille. Using a set of commands (essentially, keyboard shortcuts or touchscreen gestures), users can navigate and interact with applications and their content (Nicolau & Montague, 2019: 321-322).

A SR provides information not only on textual information but on the meaningful functions and relationships within and among other semiotic resources, such as images, icons, and navigation menus; structural elements, such as headers, footers, lists or tables; and the affordances (Torres-del-Rey & Morado Vázquez, 2019: 2-3) and results of interacting with digital objects (e.g. whether there is a block of information that can be expanded or collapsed, what state it is in, and when the user activates it). It can also advise the user on how to interact with those digital objects.

[2] See https://scholar.google.com, https://www.scopus.com, and https://webofscience.com.

Therefore, SRs crucially mediate human-computer interaction (HCI). And they do so, so to speak, by enacting or "giving voice to" the bi-directional, multi-modal conversation (Podmajersky, 2019: 53-61) that users have with digital content when they browse and interact with it. This usually takes place in auditory or braille form from SRs to users, and through keyboard action or voice commands from users to SRs. In this article, we will only focus on voice synthesis and keyboard interaction, and we will not consider other technologies that may integrate with SRs, such as screen magnifiers or braille displays.

A particularly challenging part of this conversation is the interplay between and among the various *human languages* of the different pieces of software involved, of the content and mechanisms the users wish to interact with, and of users.

Language — the words and syntax used to describe the experience that users have with a device — is one of the most important elements in the user journey, conversation, or interaction with the digital content. Design, as the process of planning and arranging the potential interactions and interfaces between digital products and persons to cater for their needs (Winograd, 1996: xv-xxv; Norman, 2013: 27-30), "is often perceived as visual, but a digital product relies on [textual] language" to a great extent (Metts & Welfle, 2020: x). By writing the words in the interactive components of a digital product, we are designing the experience someone has with it, to help them accomplish their tasks (ibid.).

In fact, "when it comes to certain types of interfaces, words are all there is" (ibid.: 10-14). That is why the Web Content Accessibility Guidelines 2.2 (Campbell et al., 2023) stress the importance of accurate, concise textual language, and of declaring the language of web pages and parts of pages, so that "[u]sers with disabilities who know all the languages used in the Web page will be better able to understand the content when each passage is rendered appropriately" by means of speech synthesisers, refreshable braille displays, and with the help of online dictionaries, translation, and other technologies (ibid.).

2. Objectives

Encouraged by García's (2020) experiments, we decided to delve into the problem of the multilingual performance of SRs from an academic perspective. In contrast with García (ibid.), we were not primarily interested in grading how well or badly each SR communicated certain multilingual HTML structural elements, labels (the actual visible text) and attributes (usually, non-visual properties of certain elements). Instead, we aimed at taking a few steps back as well as moving forward.

Stepping back to gain perspective meant that we needed to understand *how* SRs interact not only with the content but also with the software involved (browsers and speech synthesisers, mainly), how they can be set up to work in a multilingual context, and *why* SRs produced speech output in one language or another for each information block. It is a move towards semiotic engineering and the analysis of metacommunication in software design and interaction, about focusing on design goals and how they are communicated (de Souza & Leitão, 2009).

Moving forward meant shifting the focus from a sole concern with whether switching between languages took place in a timely and correct manner to the actual multilingual *utterances*. Particularly, we were interested in identifying what kind of information was produced in the HCI of SRs, as well as in the accuracy and effectiveness of the translations of those utterances.

We also grew uncomfortable about the linear, static nature of the exploration of the multilingual rendering of SRs in García's study. This is because one of the research principles in our approach to accessibility is that reading and using a digital product is essentially dynamic and conversational in nature (Podmajersky, 2019: ch. 3; Winograd, 1996), based on specific purposes, interactions, and semiotic exchanges of meaning-making.

With the above objectives in mind, we decided that we needed an essentially qualitative — rather than quantitative — approach to deal with the two main sources of data we set out to gather: on the one hand, the results of interaction with digital information in a multilingual setting through SRs; and, on the other hand, García's article itself. Given our purposes and our mixed data sources, we opted for thematic coding (Saldana, 2021) to annotate the phenomena under study, and borrowed a useful qualitative construct from a recent study (De la Cova & Torres-del-Rey, 2024): the elicitation of translation or localisation problem categories, which would only emerge, following Grounded Theory (Corbin & Strauss, 2008), after letting "data speak for itself" (Glaser & Holton, 2004: 4, in De la Cova & Torres-del-Rey, 2024: 170).

Before explaining our method of analysis, we will provide a summary of how SRs work from a technical point of view. We will also present García's (2020) experiments, his argumentation and his discussion of results, which, as mentioned, will serve as a basis for our own, enhanced study. Both summaries will be instrumental in understanding our methodology and the subsequent discussion of our results.

3. Technical foundations of multilingual interaction by screen readers

The process of synthesising speech from browser interaction with a SR (with most processes completing within milliseconds) is more complex than can initially be assumed (Roselli, 2023), and it requires a careful examination of the different agents and technical layers involved.

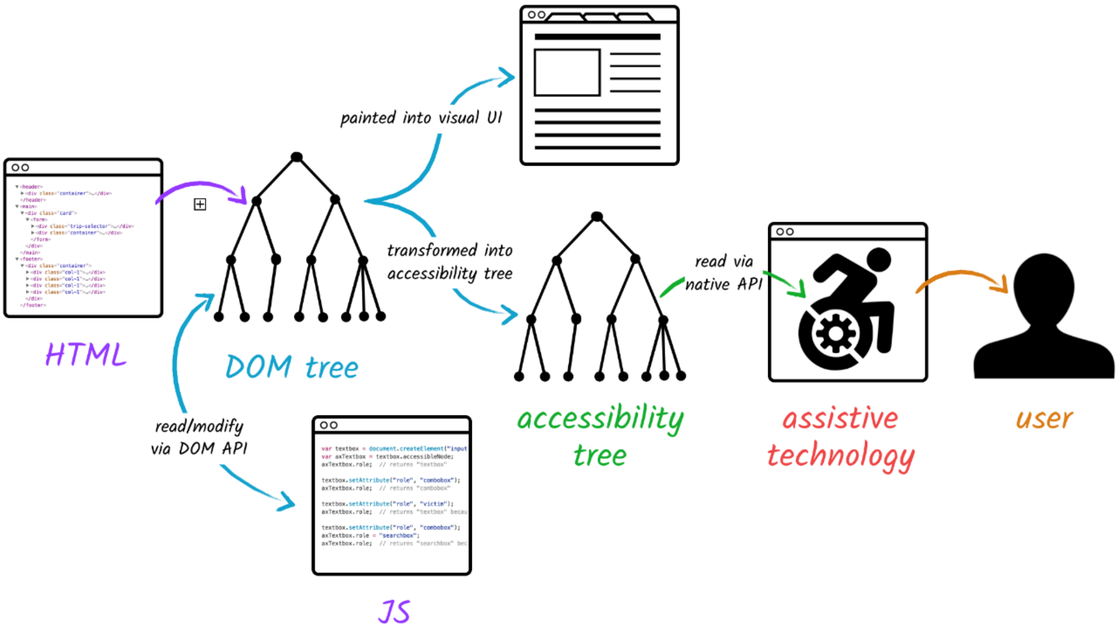

Figure 1. Agents and technical layers in SRs' multilingual interaction.[3]

A diagram of how this process works is provided in Figure 1. Initially, web developers use markup languages like HTML and SVG, along with ARIA attributes (Nurthen, Cooper & Henry, 2024), if needed, to define content structure and semantics. Browsers translate and map this markup into the Document Object Model (DOM), described by the W3C as "a platform- and language-neutral interface" (Apparao et al., 1997). The DOM reflects the webpage's information and organises it into a structured tree of objects (WHATWG, 2024a).

Modern web browsers, such as Mozilla Firefox, Google Chrome or Microsoft Edge, generate a subset of the DOM known as the accessibility tree (ACT). The ACT is "a hierarchical representation of elements in a UI or document, as computed for an accessibility API [...] a separate, parallel structure to the DOM tree" (Richards, 2019). In the ACT, semantic HTML elements like headings, lists, and links provide critical information to SRs about the content and its organisation. Conversely, some elements may be excluded if they are hidden or lack semantic relevance (ibid.)

The hierarchy of objects from the DOM or ACT is then exposed to assistive technologies (ATs), such as SRs. These technologies parse the webpage code to process accessibility-related data, capturing details like the name, description, role, and state of most HTML elements. This data is typically converted into natural language (e.g. "heading level 2" for an `h2` element) before being forwarded for processing to either the operating system (OS) or the SR's text-to-speech (TTS) synthesiser, generally using male or female synthetic voices to utter the text. One of the key properties that can be exposed by the browser through the ACT is the language of each specific element and attribute.

[3] From Boxhall et al. (2023). Copyright © 2017-2024. World Wide Web Consortium. https://www.w3.org/copyright/software-license-2023.

Finally, users do not simply listen to the synthesised audio output: they actively engage in the conversational experience using keyboard commands to navigate and interact with websites. When elements allow for interaction, users' commands are reflected on the user interface (UI), which then triggers modifications to the DOM and ACT that SRs then capture and interpret in a loop. Considering the major role SRs play in this interaction, they are key in (mis)leading users in their conversational journey with(in) web pages.

TTS synthesisers are pivotal components within SR software, tasked with transforming textual content into audible speech. These sophisticated systems use advanced algorithms and linguistic models to interpret written text and generate lifelike vocal output (Taylor, 2009), usually known as (synthetic or synthesiser) voices. Contrary to common misconception (TPGi, 2017), the responsibility for generating audio output lies with the speech synthesiser, not the SR itself. Consequently, all SRs come equipped with a built-in TTS system, such as Eloquence for JAWS or eSpeak for NVDA. OS synthesisers can also be used, but some do not have language-switching capabilities.

SR users can download additional TTS synthesiser drivers or plugins, and language-specific voices for those synthesisers (from providers such as Acapela, Nuance, and IVONA). These often offer higher quality and more variety, and significantly improve speech intelligibility and naturalness through advancements in prosody, intonation, and adjustable settings for pitch, speed, and volume. From customising preferences within OSs, browsers, and SRs to selecting and downloading voices, users can play an active role in tailoring their own digital experience.

4. The research groundwork: García's study

As mentioned in the Introduction, García's (2020) was a pioneering article. By using a large combination of SRs and browsers (both for desktop computers and mobile phones), he was able to determine that the "state of multilanguage support at the interface between all major SRs and browsers was truly appalling"; he wondered, "how is it possible that we're in the year 2020 and having all these issues?"

His main objective was to identify "perfect", adequate, incorrect, or plainly "wrong" multilingual speech synthesis by SRs when reading web content. While taking his results very much into account, we have shifted our focus towards a more qualitative and comprehensive approach, as will be explained in more detail later.

First, however, we decided to scrutinise García's methodology, assumptions, results, and discussion to determine what we could re-use, and to identify emerging themes that were worthy of further consideration in our qualitative study.

One of the interesting decisions that García made was to include different SR reading modes (*cfr.* Access for All, 2020), which we expanded, as will be seen in section 5.3:

1. *Continuous reading*, where the SR reads continuously from top to bottom (or from a specific point). This is useful particularly in short pages, the first time a page is encountered, or for descriptive parts of the page.
2. *Keyboard navigation*. This has several variations. García mentions "keyboard shortcuts", which sometimes involve pressing a special key for each SR (the "activation key") plus another key representing the element to be read (e.g. "H" for headings, or "G" for graphics). However, in García's data repository, the only action registered is using the keyboard arrows to move the reading cursor to the next unit (typically, a line of text), or, in the case of VoiceOver for Mac, its activation key plus the right arrow, which moves to the next HTML element.
3. *Tabbing*, or focus mode, where interactive elements (hyperlinks, buttons, form fields) are focused[4] consecutively by means of the tab key. García only used it on experiment 2 (of 3).
4. *Swiping* in touch phones, with a similar behaviour to keyboard navigation.

García's assessment of multilingual speech synthesis was mainly based on the following parameters:

1. Whether **textual content** other than element names (e.g. "Related information" and not "heading level 3" for an `h3` element containing that text; the alternative text description for an image; or a simple paragraph) was synthesised in a **voice** whose language (which we will call the "**SynthVoiceLang**") matched the language established by the `lang` attribute for that content.
2. Whether the names of HTML page elements (parts of the structure or interface like header or footer, objects like images or links, or the level 3 heading in the previous example) were produced in the language set for the UI of the SR (the "**SRUILang**", henceforth) or in the textual content language (the language established by the `lang` attribute for that content). Thus, the SR could send `<button lang="es">` to the synthesiser as *button* if the SRUILang is English or as *botón* following the language of its content. García does not consider one more correct than the other, provided that the language of element names matches the SynthVoiceLang (*button* being pronounced in an English voice or SynthVoiceLang, or *botón* in Spanish).
3. Whether there was repetition of element descriptions or textual labels, additional content, an element skipped when it should not have been, or not skipped when it should have been (e.g. a decorative image with null alternative text meant to be ignored by the SR). As this does not specifically concern the multilingual aspect under study, we ignored such issues.

The SR exploration by García (2020) was limited to the following elements and their textual labels on an English-language page (`<html lang="en">`), with a few items in Spanish:

- Main navigational or structural elements: page title, top-level headings, paragraphs, and ordered lists with several items.
- Graphical elements: images with an alternative text, one of them with an explicit `lang="en"` attribute, and an image with a null alternative text.
- Interactive elements: links and buttons.

[4] On keyboard focus, see https://web.dev/learn/accessibility/focus.

- Various alternative text techniques for the close "X" symbol on buttons, including `aria-label` and `aria-labelledby` attributes,[5] visually hidden text and SR-hidden (`aria-hidden`) text.

An interesting aspect of García's experiment is that the OS, SR, and browser UIs were all in Finnish.

5. Enhanced test pages and methodology

As we were interested in analysing not only the speech synthesis of common HTML elements and attributes with a variety of labelling methods in a multilingual setting, but also more comprehensive, dynamic use situations and multilingual aspects, we decided to include a wider range of elements, attributes, and interaction situations.

In this process, we took into account three specific use cases that are typical in multilingual situations: (i) computer-assisted translation (CAT) tools; (ii) online multilingual dictionaries; and (iii) cultural-specific aspects that are bound to cause problems in localisation and multilingual content, such as dates, currencies, punctuation, and so on. We consulted a few online CAT tools and online dictionaries to come up with typical structures, elements and conversation items, and we also checked a list of international variables (García Nevares, 2016: 67-69). In 5.2, we will specify which items we included in relation to each of the use cases (i, ii, or iii). We also made sure that some of these new items had a higher level or interactivity as compared to García's (2020), resulting in additional status feedback utterances by the SRs, which will be indicated as "iv".

To ensure the study remained within a manageable scope, rather than analysing the above aspects *in vivo* we decided to transfer them to the two baseline test pages by García, adding a new repository in GitHub. As mentioned earlier, we also decided to use not only linear navigation, whether continuous, keyboard or tab-based, but to employ more interactive, non-linear methods of navigation. Also, we sometimes made ad-hoc changes to different multilingual settings to corroborate the patterns of behaviour we had detected.

5.1 Screen readers, browsers, synthesisers, and languages

The choice of SR and browser combinations was made following the most recent SR usage survey (WebAIM, 2024), discarding JAWS with Firefox due to faulty behaviour. We only used one browser with TalkBack for Android and VoiceOver for iOS and for Mac, as those are the overwhelmingly most frequent combinations for these SRs. Here is the full list of the combinations we used:

- **JAWS** with Chrome and Edge (Windows). Vocalizer Expressive synthesiser.
- **NVDA** with Chrome and Firefox (Windows). Vocalizer Expressive synthesiser.
- **VoiceOver** with Safari, for Mac OS (**VO-Mac**) and iOS (**VO-iOS**). Built-in synthesisers.

[5] See https://tink.uk/the-difference-between-aria-label-and-aria-labelledby about these ARIA attributes.

- **TalkBack** with Chrome (Android). Samsung TTS synthesiser.[6]

All combinations were made with two different settings, where the browser UI, SRUILang, and initial synthesiser voice (SynthVoiceLang) were all either in English or in Spanish. These are natural configurations for multilingual users. In VoiceOver (both desktop and mobile) and TalkBack, languages also needed to be matched by the OS language to avoid distorted outputs. In all, we had 14 different combinations of SR, browser, and SRUILang.

5.2 Tested content

To the content described in section 4, we added the following multilingual elements and attributes (of the types specified in parentheses, as explained earlier) on each test page:

On test page 1:

- A new button with Spanish content "Soy un botón" (i, ii).
- Several `time` elements with and without `datetime` attributes for precise dates, times and time durations. These elements were placed at and on standard numerical times and dates, as well as on a specific celebration day (Valentine's Day). Some date elements were given added "en-US", "en-GB", or "es" `lang` attributes. (i, iii)
- Other local variables: phones, US and EUR currencies, measurements, and numbers (millions and billions). (i, ii, iii)
- Abbreviations with `title` attributes, in a third language (German). (ii)
- Incorrectly written (wrongly accented) Spanish words. (i)
- Various Spanish and English punctuation marks. (i, ii, iii)
- Links on several of the above elements, some with `title` attributes and one with an attribute to open a new window (`target="_blank"`). (ii, iv)

On test page 2:

- Several page landmarks (`header`, `footer`, `main`, `aside`, `nav`), including `aria-label` attributes to distinguish same-type landmarks for the two `navs` (one a main navigation menu, the other a breadcrumb navigation). (i, ii)
- Several headings of different levels, one of which ("Fuente de datos") is visually hidden in the footer. (i, ii)
- An image with a relevant `alt` text as part of a linked sentence. (iv)
- A same-page link. (iv)
- An accessible data table with a caption, and data in different languages (simulating a CAT editor table). (i, ii, iv)
- A collapsed `details` element with a clickable `summary` to expand the information inside. (i, ii, iv)
- A definition list. (ii)

5.3 Browsing and activation methods

Besides continuous navigation, we used linear keyboard and tabbing navigation for desktop browsers and linear default swiping navigation for touch smartphone browsers.

[6] JAWS: https://freedomscientific.com/products/software/jaws; NVDA: https://www.nvaccess.org; VoiceOver: https://www.apple.com/accessibility/vision; TalkBack: https://support.google.com/accessibility/android/topic/3529932.

In the case of NVDA, JAWS (with up/down keyboard arrows) and smartphone default swiping, navigation tends to be line by line. VO-Mac has this navigation mode too, but we also used the combination of its activation key (Control+Option, or Caps Lock) and right/left arrows to navigate by HTML element, which is typical for this SR.

We used navigation by specific elements extensively (headings, landmarks, links, lists, images, tables), by means of key combinations in desktop browsers and by means of navigation mode selection in mobile phones, including the "rotor" in VoiceOver (for both iOS and Mac OS).[7]

Tabbing allowed us to select interactive elements purposefully, which, as we will see, provides better UX (user experience) and accessibility in terms of language synthesis than plain, line-by-line, up/down arrow navigation.

Whenever possible, we activated interactive elements, such as links or a disclosable `details` element, which caused state changes (visited link, collapsed or expanded toggle button).

5.4 Speech synthesis, transcription, and coding strategy

SRs were set at their default configuration, including verbosity level (i.e. the details provided about the content being spoken, like punctuation, type of element, or region the content is in, and so on).

We developed an MS Word table to transcribe the SR voice synthesis, with the following columns:

1. Utterance number.
2. Transcribed speech synthesis.
3. SynthVoiceLang for textual content.
4. SynthVoiceLang for element names.
5. HTML code.
6. Comments, including speech syntheses in non-continuous reading modes.

We first used NVDA's speech viewer tool, which output most of the content synthesised by that SR into text format. To help us manually complete the NVDA transcription and make any necessary changes in the transcription of other SRs' output, we recorded the speech syntheses with the following audio and screen capturing software: Audacity and Camtasia for Windows, Quicktime for Mac, ScreenPal for Android, and the built-in screen-recording feature for iOS.

In column 1, each row roughly corresponds to a specific line utterance, an element description, or both if related. It was sometimes necessary to subdivide rows to avoid overcrowding them with too many annotations.

[7] See https://www.afb.org/blindness-and-low-vision/using-technology/cell-phones-tablets-mobile/apple-ios-iphone-and-ipad-2.

For column 2, the following transcription conventions were established:

- Textual content was transcribed in **bold**, and element descriptions in *italics*.
- Where there was a mismatch between the SynthVoiceLang and the content language (whether the textual content, the element name or its description), we marked that specific utterance in high-contrast red font and asterisked it.
- When we thought a language mismatch was noteworthy (e.g. because it could be considered legitimate), we highlighted it in cyan colour.
- When transcribing the pronunciation of certain items was deemed necessary (e.g. numbers, punctuation, or mispronunciations), the item was immediately followed by the pronunciation in square brackets, and both the item and pronunciation transcriptions were highlighted in yellow, or green to distinguish consecutive items.
- When certain visual written output items were not audio synthesised (e.g. punctuation marks), they were highlighted in pink.
- Some small element descriptions that were only uttered in non-continuous mode were included in square brackets.

For columns 3 and 4, the same conventions were followed in terms of the use of bold (column 3), italics (column 4), and red font plus asterisks and cyan highlighting (both columns). Capitalised ISO language codes were used (EN, ES, EN-GB, EN-US, DE); to further distinguish EN from ES, we underlined the latter ("ES").

In column 6, we included our comments, as well as long element descriptions or interaction feedback uttered in specific keyboard/swiping, tabbing, and element-by-element navigations.

6. Results

Screen-reading the contents of both test pages with the above settings resulted in the following categories of problems related to translation and multilingual accessibility. Rather than considering these problems as bugs (although some may be), we placed our focus on translation and localisation aspects, as well as hypothetical design goals, assumptions about user needs, or the results of constraints in design, in line with the semiotic engineering approach mentioned in section 2.

6.1 Linguistic aspects

These aspects are related to the linguistic form of webpage interface and content utterances.

6.1.1 Terminology and morphology

While some elements are translated consistently across SRs (e.g. *tabla* for the "table" element, *enlace* for a "link", *navegación* for the "navigation" landmark/region, *fila* for a "row" in a table, and *página actual* for the "current page" attribute of a navigation menu link), others show interesting variations between SRs, as seen in Table 1. This may confuse users switching from one SR to another.

It is worth noting that landmark/region names are sometimes based on the HTML 5 sectioning element name (e.g. `header`, `footer`, `aside`) or the role name given as an attribute in the WAI-ARIA accessibility specification (`banner`, `contentinfo`, `complementary` respectively) (Faulkner, 2013).

Element or state	JAWS	NVDA	VoiceOver	TalkBack
<aside>	*Información complementaria* (complementary information) *Información adicional* (complementary region)	*Complementario* (complementary)	*Complementario* (complementary)	*Complementario* (complementary)
<footer>	*Información adicional (content information)* *información complementaria* (contentinfo region)	*Información de contenido* (content info)	*Información del contenido* (content information)	*Pie de página* (footer)
<header>	*Título* (banner)	*Título* (banner)	*Tira* (banner)	*Báner* (banner)
heading level X (`<h1>`, `<h2>`...)	*Nivel de encabezado X* (heading level X)	*Encabezado nivel X* (heading level X)	*Encabezamiento de nivel X* (heading level X)	*Encabezado X* (heading X)
Toggle button states	*Contraído/ Expandido* (collapsed/ expanded)	*Contraído/ Expandido* (collapsed/ expanded)	*Contraído/ Ampliado* (collapsed/ expanded)	*Oculta/ Mostrado* (collapsed/ expanded)

Table 1. Translation of element and state names in the different SRs.

The most striking translation choice is for the `header` landmark. *Báner* (TalkBack) is not a common Spanish word except in the case of ads on web pages. *Título* (JAWS and NVDA) is confusing because it is the usual translation for the `title` element and typically refers to a sentence or a longer phrase, not to a region. But the oddest translation is *tira* (VoiceOver), which conjures up images of a cloth or a comic strip. It may have originated from one of the dictionary equivalents of banner as a kind of heraldic flag.

The Spanish translation of JAWS mixes up the `aside` and the `footer` landmarks. This SR announces the former in English as "complementary information" when browsing by landmark, but as "complementary region" when entering that region otherwise (no doubt to avoid a very long landmark name). Similarly, the latter is read as "content information" and "contentinfo region" respectively. This may have caused the mix-up in Spanish, as the former is translated as *Información complementaria* (complementary info) when browsing by landmark and as *Información adicional* (additional info) when entering otherwise; but the same translations are given for the latter landmark, albeit in the reverse contexts.

Finally, TalkBack uses a different grammatical gender in Spanish for the “collapsed” (*oculta*) and “expanded” (*mostrado*) toggle button states.

6.1.2 Syntax

One of the salient characteristics of the translation of the type of elements and their descriptions is the syntactic articulation of adjective-noun groups, in several specific contexts:

1. The type of landmark or region. Most SRs translate *type of region* + “region/landmark” without switching to the NOUN+ADJ order in Spanish. Examples: *navegación region*, *título región* (NVDA), *información del contenido punto de referencia* (for “contentinfo landmark”, VoiceOver).
2. The distinctive label given by the author to a landmark or region (when there are other same-type landmarks). Example: “Main *navigation region*” is produced in Spanish SRUILang as “*Main* *región navegación*” (JAWS) for `<nav aria-label="Main">`.
3. The state (or pseudo-class) of certain elements. Example: “*visited, link*” is announced as “*visitado, enlace*”. Compare with another case in JAWS where, after browsing by links to an `h3` element, the SR produces “*Nivel de encabezado 3, enlace a esta página, visitado*” (`<h3><a href="#footer">`).

6.1.3 Pronunciation of unknown words

All SRs pronounced words with erroneously accented characters as written, that is, with wrongly stressed syllables.

Synthesisers were often successful in pronouncing (that is not to say *intoning*) common foreign words, particularly in English (with Spanish voices). This suggests some English-centric bias, which would merit further research.

6.2 Locale-specific aspects

The following aspects can usually be configured in the SR to be processed in different ways or to be controlled by the synthesiser.

6.2.1 Punctuation

SRs (particularly desktop SRs) have user-definable settings for punctuation verbosity and to decide whether punctuation marks are sent to the synthesiser as words directly (for instance, “dot”, “period” or “full stop” for “.”) or as symbols (to be converted into a word by the synthesiser). Default settings may vary by language.

Concerning verbosity, punctuation can either be integrated in the spoken sentence (thus not pronounced, although possibly expressed through intonation) or spelled out. In JAWS, we have found that this behaviour usually depends on the synthesised language: punctuation verbosity is high for English content (hyphen, colon, left quote, full stop, greater than, and so on), but low for Spanish.

As for the language in which punctuation is expressed, it can be either the content language (JAWS, NVDA, VO-Mac) or the SRUILang (TalkBack). It remains to be seen which approach is preferred by users. However, it seems reasonable to suggest that switching languages for punctuation in the middle of a line or a sentence may act as an element of distraction, but not so much at the end. What is certainly a problem is to have a punctuation mark converted to a word in the SRUILang but pronounced in a different SynthVoiceLang (TalkBack).

6.2.2 Abbreviations

Substitution strings are typically defined in language-specific SR dictionaries (sometimes provided by the installed synthesisers) for common acronyms and abbreviations (e.g. “et cetera” for “etc.”), or otherwise processed directly by SRs by means of their own dictionaries.

An interesting problem is shown in the translation of the JAWS UI into Spanish: when setting the primary SR language, one can choose the OS language as default, which is translated on the UI as “Idioma del SO” but announced as “Idioma del sudoeste” (back-translation: “Southwest Language”).

A reasonable solution here would be to code abbreviations with their full meaning. This is one of the sufficient techniques for abbreviations in WCAG 2.2 (`<abbr>` element with `title` attribute). However, SRs usually do not announce that attribute by default, nor do they provide easy mechanisms for producing both abbreviation and full form (an exception in the latter case is VO-Mac, when browsing by line).

6.2.3 Calendar elements

Times and dates are usually guessed by SRs or synthesisers, meaning that certain text formats are identified as calendar elements (e.g. dd-mm-yy in NVDA, whether in English or Spanish) and announced following locale-specific date or time formats.

In contrast to the above guessing strategy, the `<date>` element is not normally parsed by SRs, so it is not used to attempt to announce a proper date or time format. One exception is VoiceOver, which synthesises the `datetime` attribute (after the guessed textual date or time in the content language) when navigating by element, and it always does so in the SRUILang and matching SynthVoiceLang.

An interesting locale-specific finding is that changing the `lang` attribute between “en-GB” and “en-US” does make a difference in the way dates are announced (day-month or month-day) by the British or the American SynthVoiceLang respectively.

6.2.4 Numbers, currencies, and measurements

Currencies are usually well announced by NVDA when using standard punctuation for whole-number thousands, millions, and billions in English and Spanish — even with a non-breaking space for thousands in Spanish, as recommended by language authorities;

this, however, is not recognised for Spanish in the Unicode CLDR, causing other SRs to announce the individual digit before the space first, and then the hundreds after a pause.[8]

Other SRs show some unexpected behaviour, such as announcing the dollar currency before the number in JAWS; and pronouncing a billion in Spanish ("mil millones") as "un millón cero cero cero" (one million zero zero zero) in JAWS (recently fixed), "un millón miles" (one million thousands) in TalkBack, or number by number and dot by dot in VoiceOver.

Units and decimals in measurements are well announced in all SRs except VO-iOS. In this SR, "dot" is used instead of "point" for decimals in English; and the SRUILang is always used for this punctuation mark, resulting in an English-pronounced "punto" in the middle of measurements in English when Spanish is set as the SRUILang.

6.3 Pragmatic aspects

6.3.1 Intermodal coherence

The names of elements as announced by SRs should be consistent with the names of those elements as specified in each SR's settings and documentation. However, this is beyond the scope of this study.

6.3.2 Commands

For keyboard shortcuts, the first letter of English words is used (alone or in combination with other keys), which may be of less mnemonic value to users of other languages. Nonetheless, most SRs offer the option of changing such keys.

This is the typical list of single-key commands, reflecting the English language primarily:

- H for headings
- L for lists (except VO-Mac) or links (VO-Mac)
- X for lists (VO-Mac)
- G for graphics
- D (NVDA) or R (JAWS) for landmarks/regions.
- F for forms (but J for form controls in VO-Mac)
- T for tables

6.3.3 Elision

Interesting elision techniques were found in our experiments due to the contiguous repetition of the same utterance in mobile SRs.

[8] Article by Fundéu Spanish Foundation for Urgent Spanish: https://www.fundeu.es/recomendacion/miles-y-millones-claves-de-escritura. Spanish number chart by the Unicode Common Locale Data Repository (CLDR): https://unicode.org/cldr/charts/45/verify/numbers/es.html.

In TalkBack, this happens with the combination of textual content and the description of the `aria-current="page"` property. So, the HTML string `<nav lang="es"…>…<a href="#" aria-current="page">Página actual</a>` produces "Página actual, *enlace*" with Spanish SRUILang, but "*Current page*, Página actual, *link*" with English SRUILang.

In VO-iOS with English SRUILang, the string `<button>I am a button</button>` produces "I am a , *button*", whereas `<button lang="es">Soy un botón</button>` gives "Soy un botón, *button*". However, this elision technique does not happen with Spanish SRUILang, resulting in "Soy un botón, *botón*".

6.4 Design and technical factors influencing language switching

These concern design choices or technical constraints that determine the language used to generate textual interface descriptions and the selection of a SynthVoiceLang.

The only relevant difference in language utterances between browsers was found in NVDA. Firefox seemed to expose more properties than Chrome, as will be seen in 6.4.3.1 and 6.4.3.2.

6.4.1 Reading mode

The selected reading mode had a significant effect on language switching, especially for desktop SRs, which are more verbose by default and tend to follow `lang` attributes less strictly in continuous mode. This is probably because sending a longer list of textual utterances without language switching orders would result in fewer synthesiser delays.

It is debatable (and subject to user studies) whether, which and to what extent language mismatches between the content and the SynthVoiceLang in continuous mode may have a negative effect on users who just want to get the gist of the content without too much language switching. However, except for complex structures and punctuation, TalkBack performed seamlessly, with appropriate language switching in continuous mode.

Element by element (in VO-Mac) and element selection modes (through the rotor in VoiceOver, three-finger vertical swiping in TalkBack, and key commands in NVDA, JAWS, and VO-Mac) showed much better performance. This is because each element is entered individually or each element type is called up specifically by the SR, so the SRUILang element name is bound to be announced in the SynthVoiceLang, rather than being "carried over" by the content language in each line of text (see section 6.4.3.1).

6.4.2 Primary language (SRUILang)

Designers seem to apply the rule that users who have the browser, the SR UI, and the synthesiser voice set to a particular language would prefer to have elements, states, and properties announced in their (primary) language (or SRUILang). This creates a multilingual and multi-voice effect when reading content that is in a language other than the SRUILang.

It would be interesting to investigate whether users prefer an option to stick to the same language for the textual content of the page and for reading element and interaction descriptions. This appears to be the intended voice strategy of VO-iOS (e.g. "Lemma, *description list start*, *term*, Casa, *descripción*" for <dl><dt>Lemma</dt><dd lang="es">Casa</dd> in either SRUILang), although it sometimes results in a great deal of language mix-up and mismatches (e.g. "Status, **encabezado de columna**, *row 1, column 3*", with everything in an English SynthVoiceLang for the string <th scope="col">Status</th>).

6.4.3 Syntactic relationships

The following patterns of syntactic relationships among elements and between elements and their attributes or properties, as well as action states, were identified as being responsible for the (correct or incorrect) choice of SynthVoiceLang.

6.4.3.1 Carry-over effect

End-of-sentence elements were more likely to be announced in the wrong SynthVoiceLang in continuous mode, linear keyboard navigation, or as a result of interaction with state changes, probably for the reasons stated in 6.4.1. Examples:

- JAWS (English SRUILang): Spanish voice pronunciation for **button** after the Spanish text shown in the button, and of **graphic** after its alternative text with lang="es". (Spanish SRUILang): English voice pronunciation for **botón contraído** after the summary text in a disclosure details element.
- VO-Mac: (English SRUILang): "Usted está aquí *navigation* *Experimento* > [greater than]" with English SynthVoiceLang for "Experimento", in string <nav aria-label="Usted está aquí" lang="es">Experimento >.
- NVDA with Firefox (Spanish SRUILang): "*triángulo relleno que apunta hacia abajo*, *Translation of word*, *expandido*" with Spanish SynthVoiceLang for English content "Translation of word" after interacting with the disclosure element to open it (<details><summary>Translation of Word</summary>).

VO-iOS behaved rather erratically. Element names were usually pronounced in the content language — being synthesised in that language — only when it was the last line item. In other line positions, element descriptions were announced in the SynthVoiceLang of the last element in the previous line.

6.4.3.2 Landmark descriptions and other HTML element attributes

In all navigation modes, the aria-label attribute and the label referenced by aria-labelledby in landmarks were understood by JAWS and NVDA as descriptive adjectives for the landmark names. This meant, in our example from the previous section, that for navigation breadcrumb landmarks labelled in Spanish, these SRs concatenated "Usted está aquí" before the landmark name, in the ADJ+NOUN order ("*navegación región*" in NVDA, and "*region navegación*" in JAWS, in Spanish SynthVoiceLang). With English SRUILang, there was a notable language mismatch, announcing the whole concatenated string ("*Usted está aquí*, *navigation landmark/region*") in English pronunciation and intonation, whereas the textual content in that navigation region, "Experimento", was correctly pronounced in Spanish SynthVoiceLang.

NVDA and VO-Mac seemed to do the same for other attributes (the `title` global attribute, the `alt` attribute for images, and the `aria-label` and the `aria-labelledby` attributes — the latter irrespective of the language of the referenced element — for interactive objects such as buttons). Example: *Cerrar* in an English voice for `<button lang="es" aria-label="Cerrar"><span aria-hidden="true"> X </span> </button>` with English SRUILang (this, however, did not happen in NVDA with Firefox, where the language change *was* exposed for the accessible name of the button). The other SRs matched the voice to the content language for these attributes, except TalkBack for `title`.

Both VoiceOver and TalkBack used the SynthVoiceLang matching the landmark content language. In the above breadcrumbs example, Spanish would be used throughout that `<nav>` landmark, for the element name, description, and the text in it.

6.4.3.3 States and properties

All SRs announced the expanded or collapsed state of their toggle element in the SRUILang and matching SynthVoiceLang, with the following exceptions: JAWS produced a mismatched SynthVoiceLang in continuous mode only; and VO-iOS always announced the state in the language and voice of the content language (*Collapsed/Expanded* plus instructions for interacting with the details toggle button in Spanish, but all in an English SynthVoiceLang).

6.4.3.4 Complex hierarchical structures

We have found some issues with certain nested structures or combinations of attributes and text nodes in elements:

- The `lang` attribute of certain generic elements, such as `<span>`, is sometimes overridden by an immediate parent element like `<a>` (TalkBack, VO-Mac).
- When the language of a hyperlink is different from the language of a child element (like `span`, `abbr`), the `title` attribute of the hyperlink is announced in the voice of the child element (JAWS, VoiceOver).
- Tables are complex structures, particularly when announcing the caption and the row or column headers associated with each data cell. When landing on a table, the `caption` is usually announced in the SRUILang, even if it contradicts that element's `lang` attribute (JAWS, VO-Mac, NVDA — although this SR then reads the caption again in the matching voice). On the other hand, the name of the header of a data cell was announced in the language of the data cell (JAWS), even if the language of the header was different; or in the SynthVoiceLang (remaining SRs).

6.4.4 Automatic language detection

Some instances of automatic language detection were identified, notably in VO-Mac. Others can be set up, as in JAWS. In the multilingual table in our experiment, including common Spanish or French words without adding a matching `lang` attribute usually produced the correct Spanish or French SynthVoiceLang.

7. Conclusions

We have shown that SRs are able to deal with multilingual documents, although sometimes with major mismatches between the language of the text to be synthesised (either content language or element descriptions) and the language of the synthesiser voice pronouncing it, particularly when reading in continuous mode. Users may also become confused when page regions and interactive elements are labelled with attributes in a language other than the one set for their SR, announcing the element name and label in mixed languages.

This article was based on García's (2020) observations and findings on the performance of SRs with multilingual content. However, we broadened the focus to try to account for the different "actors and networks involved" (Torres-del-Rey, 2022: 154, following Latour), both human and technological, including designers, users, translators, browsers, and TTS synthesisers. In this sense, we considered users' interaction with the digital content through the SR as a multi-directional, multimodal conversation. In future analyses, it would also be important to ask users specifically to include other semiotic resources such as pauses and non-verbal sounds, and to develop advanced transcription methods for conversational interactions, which could be inspired by multimodal conversation analysis (Jewitt, Bezemer & O'Halloran, 2016: ch. 5).

We also wanted to approach our research object from a broader, translation-oriented perspective, adopting qualitative analysis tools inspired by semiotic engineering (de Souza & Leitão, 2009) and Grounded Theory (De la Cova & Torres-del-Rey, 2024). This is probably our main contribution: a set of categories and subcategories of translation and localisation phenomena (linguistic, locale-specific, pragmatic, and design-technical aspects) have been elicited through the analysis of data from our experiment. These aspects could be used as a template in future studies and tests on not only the performance of SRs with multilingual content, but also the design of SRs' multilingual interaction, and in user reception studies.

The interrelation of accessibility components (Henry & Duffy, 2024)[9] in the multilingual production of SRs could be further expanded in a critical review of the compliance of

[9] In 2004, the Web Accessibility Initiative (WAI) produced a seminal concept and illustration of the essential, interdependent components of web accessibility. A triangle with three vertices (content, developers and users) showed the interaction of these components: on the content side, accessibility guidelines and technical web specifications; on the developer side, authoring and evaluation tools; and on the user side, user agents such as browsers, media players and assistive technologies. See https://www.w3.org/Talks/wai-components/Overview.

this specific — but key — aspect of linguistic accessibility with the User Agent Accessibility Guidelines (UAAG), ergonomic approaches to accessibility in human-system interaction, such as those in the ISO 9241 series (ISO 9241-20:2021), as well as through usability heuristics such as Nielsen & Molich's,[10] to ensure that users get, for example, greater control, better error recovery, and standardised naming conventions, also in translation — including familiar, locale-specific terminology, syntax, and synthesis of punctuation, abbreviations or numbers. Online dictionaries or CAT tools could also provide a file with appropriate, context-specific conversions of punctuation, abbreviations, or other relevant information, which could be imported through the SR settings utility.

Another step towards a comprehensive approach to accessibility components for multilingual content would be to examine the specifications for the `lang` attribute in the light of documents such as CAT tools or multilingual dictionaries, where languages are organised in patterned but non-consecutive structures, such as table columns or rows. Would it not be useful to have an attribute defining the column or row *scope* of the `lang` attribute, as is the case for headers in tables? Otherwise, do authoring tools provide an easy way to replicate this attribute across an entire column or row? Other W3C specifications that could be explored for their contribution to both accessibility and localisation in HCI are MathML, for numerical expressions, and Spoken Presentation.[11]

In an increasingly globalised and AI-intensive world, it is paramount that checks and balances are in place to ensure that information and interaction mechanisms in media are accessible and locale-sensitive, to allow everyone to be connected and to participate in and contribute to a more inclusive society. For this to happen, we must get involved, from humanistic fields, in the co-design of technologies crucially mediating the relationship between humans and reality, without shying away from the technical complexity at play, which is inevitably — if covertly — laden with values and assumptions about the subjects involved (Torres-del-Rey, 2022: 153).

8. References

Access for All (2020). Browse and Focus Modes. *Accessibility Developer Guide*. <https://www.accessibility-developer-guide.com/knowledge/screen-readers/desktop/browse-focus-modes>. [Accessed: 20241219].

Apparao, Vidur et al. (1997). *Level 1 Document Object Model Specification* — Last Updated 20 July, 1998. W3C. <https://www.w3.org/TR/WD-DOM/. [Accessed: 20241219].

[10] See https://www.w3.org/TR/UAAG20, https://www.iso.org/standard/80709.html, and https://www.nngroup.com/articles/ten-usability-heuristics.

[11] For MathML, see https://www.w3.org/Math and https://w3c.github.io/mathml-docs/gap-analysis. On Spoken Presentation, see https://www.w3.org/WAI/pronunciation, https://www.w3.org/TR/spoken-html, and https://www.w3.org/TR/pronunciation-gap-analysis-and-use-cases.

Boxhall, Alice; Craig, James; Mazzoni, Dominic; Surkov, Alexander (2023 [2017]). Accessibility Object Model. *The Accessibility Object Model (AOM)*. <https://wicg.github.io/aom/explainer.html>. [Accessed: 20241219].

Campbell, Alastair; Adams, Charles; Montgomery, Rachael Bradley; (eds.) (2023). Understanding Success Criterion 3.1.2: Language of Parts. *Web Content Accessibility Guidelines (WCAG) 2.2*. W3C Recommendation 05 December 2024. <https://www.w3.org/WAI/WCAG22/Understanding/language-of-parts.html>. [Accessed: 20241219].

Casalegno, Elisa (2018). *Usability of Partially Localised Websites in Switzerland: A Study with Screen Reader Users* [Master's thesis]. Université de Genève, Geneva, Switzerland.

Corbin, Juliet; Strauss, Anselm (2008). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 3rd ed. Thousand Oaks, CA: Sage. <https://doi.org/10.4135/9781452230153>. [Accessed: 20241219].

De la Cova, Elena; Torres-del-Rey, Jesús (2024). Translation Problems in App Localisation: The Case of Google Home. In: Calvo, Elisa; De la Cova, Elena (eds.) *A Qualitative Approach to Translation Studies: Spotlighting Translation Problems*. London and New York: Routledge, pp. 166-184. <https://doi.org/10.4324/9781003253594-12>. [Accessed: 20241219].

de Souza, Clarisse S.; Leitão, Carla F. (2009). *Semiotic Engineering Methods for Scientific Research in HCI*. San Rafael, CA: Morgan & Claypool. <https://doi.org/10.1007/978-3-031-02185-5>. [Accessed: 20241219].

Faulkner, Steve (2013). Using WAI-ARIA Landmarks – 2013. *TPGi*. <https://www.tpgi.com/using-wai-aria-landmarks-2013>. [Accessed: 20241219].

García, Xurxe Toivo (2020). The troubled state of screen readers in multilingual situations. *UX Collective*. <https://uxdesign.cc/the-troubled-state-of-screen-readers-in-multilingual-situations-f6a9da4ecdf3>. [Accessed: 20241219].

García Nevares, Luis A. (2016). *Localización e internacionalización de software: puntos de encuentro entre el localizador y el programador* [PhD thesis]. Universidad de Salamanca. <https://gredos.usal.es/handle/10366/132948>. [Accessed: 20241219].

Henry, Shawn Lawton (ed.); Duffy, Michael (art.) (2024) Essential Components of Web Accessibility. *Web Accessibility Initiative (WAI)*. W3C. <https://www.w3.org/WAI/fundamentals/components>. [Accessed: 20241219].

Jewitt, Carey; Bezemer, Jeff; O'Halloran, Kay (2016). *Introducing Multimodality*. London and New York: Routledge. <https://doi.org/10.4324/9781315638027>. [Accessed: 20241219].

Metts, Michael J.; Welfle, Andy (2020). *Writing Is Designing: Words and the User Experience*. New Work: Rosenfeld Media.

Nicolau, Hugo; Montague, Kyle (2019). Assistive Technologies. In: Yesilada, Yeliz; Harper, Simon (eds.) *Web Accessibility: A Foundation for Research*. 2nd ed. London: Springer, pp. 317-336. <https://doi.org/10.1007/978-1-4471-7440-0_18>. [Accessed: 20241219].

Norman, Don (2013). *The Design of Everyday Things*. Revised and Expanded Edition. New York: Basic Books.

Nurthen, James; Cooper, Michael; Henry, Shawn Lawton (eds.) (2024). WAI-ARIA Overview. *Web Accessibility Initiative (WAI)*. W3C. <https://www.w3.org/WAI/standards-guidelines/aria>. [Accessed: 20241219].

Ong, Yoon W. (2002). Investigation of a formant-coded diphone approach to adopt new languages for multilingual speech synthesis [PhD thesis]. The University of Manchester. <https://www.proquest.com/dissertations-theses/investigation-formant-coded-diphone-approach/docview/2807860310/se-2>. [Accessed: 20241219].

Podmajersky, Torrey (2019). *Strategic Writing for UX: Drive Engagement, Conversion, and Retention with Every Word*. Sebastopol, CA: O'Reilly.

Raghavendra, E. Veera; Prahallad, Kishore (2010). A multilingual screen reader in Indian languages. *2010 National Conference On Communications (NCC)*, Chennai, India, 2010, pp. 1-5. <https://doi.org/10.1109/NCC.2010.5430191>. [Accessed: 20241219].

Richards, Melanie (2019). Semantics to Screen Readers. *A List Apart*. <https://alistapart.com/article/semantics-to-screen-readers>. [Accessed: 20241219].

Rodríguez Vázquez, Silvia (2015). Exploring Current Accessibility Challenges in the Multilingual Web for Visually-Impaired Users. In: *The 24th International World Wide Web Conference (WWW) 2015 Companion Volume.* ACM, New York, NY, pp. 871-873. <https://doi.org/10.1145/2740908.2743010>. [Accessed: 20241219].

Roselli, Adrian (2023). Don't Override Screen Reader Pronunciation. <https://adrianroselli.com/2023/04/dont-override-screen-reader-pronunciation.html>. [Accessed: 20241219].

Saldana, Johnny (2021). *The Coding Manual for Qualitative Researchers*. 4th ed. London: Sage.

Taylor, Paul (2009). *Text-To-Speech Synthesis*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511816338>. [Accessed: 20241219].

Torres-del-Rey, Jesús (2022). Machines and Technology. In: Ferreira, Aline; Schwieter, John W. (eds.) *Introduction to Translation and Interpreting Studies*. Hoboken, NJ: Wiley-Blackwell, pp. 153-188.

Torres-del-Rey, Jesús; Morado Vázquez, Lucía (2019). Transferring web accessibility through localization and internationalization standards. *The Journal of Internationalization and Localization* (JIAL), v. 6 n. 1, pp. 1-23. <https://doi.org/10.1075/jial.19002.tor>. [Accessed: 20241219].

TPGi (2017). Whose Voice is it Anyway. <https://www.tpgi.com/whose-voice-it-anyway/>. [Accessed: 20241219].

WebAIM (2020). Semantic Structure: Regions, Headings, and Lists. *WebAIM*. Institute for Disability Research, Policy, and Practice (Utah State University). <https://webaim.org/techniques/semanticstructure>. [Accessed: 20241219].

WebAIM (2024). Screen Reader User Survey #10 Results. *WebAIM*. Institute for Disability Research, Policy, and Practice (Utah State University). <https://webaim.org/projects/screenreadersurvey10/>. [Accessed: 20241219].

WHATWG (Apple, Google, Mozilla, Microsoft) (2024a). *DOM Living Standard* — Last Updated 17 December 2024. <https://dom.spec.whatwg.org/>. [Accessed: 20241219].

WHATWG (Apple, Google, Mozilla, Microsoft) (2024b). *HTML Living Standard* — Last Updated 17 December 2024. <https://html.spec.whatwg.org/>. [Accessed: 20241219].

Winograd, Terry (ed.) (1996). *Bringing Design to Software*. New York: ACM Press. <https://doi.org/10.1145/229868>. [Accessed: 20241219].