GoodIT '25: International Conference on Information Technology for Social Good
September 3 - 5, 2025
Antwerp, Belgium

Conference Sponsors:
SIGCAS

RESEARCH-ARTICLE

Toward Fully Accessible Digital Events: Guidelines, Platform Design, and PPTX-to-HTML Conversion

SALVATORE GATTO, University of Padua, Padua, PD, Italy

OMBRETTA GAGGI, University of Padua, Padua, PD, Italy

GoodIT '25: Proceedings of the 2025 International Conference on Information Technology for Social Good (September 2025)
https://doi.org/10.1145/3748699.3749815
ISBN: 9798400720895

Toward Fully Accessible Digital Events: Guidelines, Platform Design, and PPTX-to-HTML Conversion

Salvatore Gatto
Department of Mathematics
University of Padua
Padua, Italy
salvatore.gatto@math.unipd.it

Ombretta Gaggi
Department of Mathematics
University of Padua, Italy
Padua, Italy
gaggi@math.unipd.it

Abstract

With the upcoming enforcement of the European Accessibility Act (EAA) in June 2025, ensuring accessibility in digital events is no longer just a best practice but a legal and ethical obligation. This paper presents a comprehensive approach for developing platforms for accessible digital events compliant with Web Content Accessibility Guidelines (WCAG) 2.2 and accessibility best practices. We identified the main components of digital events, i.e., video player, presentation viewer, and chat, and defined specific accessibility criteria that they must meet. Then, we developed a tool that converts PowerPoint presentations into accessible HTML pages, enabling all participants, including those using screen readers, to perceive and interact with presentation content directly within the browser. Our goal is to overcome the current limitations of traditional screen sharing. We validate our converter through a series of quantitative and qualitative tests. Results confirm both high parsing accuracy and compliance with WCAG 2.2 Level AA, thus proving the effectiveness of our tool in enhancing accessibility and reducing barriers to participation in digital events.

CCS Concepts

• **Human-centered computing → Accessibility**; *Web-based interaction*; Empirical studies in accessibility; User interface design.

Keywords

Accessibility, Web Applications, Human-Centered Computing, Assistive Technologies, Web Standards, WCAG

ACM Reference Format:
Salvatore Gatto and Ombretta Gaggi. 2025. Toward Fully Accessible Digital Events: Guidelines, Platform Design, and PPTX-to-HTML Conversion. In *International Conference on Information Technology for Social Good (GoodIT '25), September 03–05, 2025, Antwerp, Belgium.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3748699.3749815

1 Introduction

Accessibility refers to the ability of all individuals, regardless of their abilities or disabilities, to interact with and benefit from products and services. In today's society, accessibility is increasingly recognised not only as a technical standard but as a fundamental human right, as affirmed by the United Nations Convention on the Rights of Persons with Disabilities (UNCRPD) [21]. According to the World Health Organisation (WHO), over 1.3 billion people worldwide live with significant disabilities that impact education, employment, and mental health [26]. Nevertheless, recent years have seen a growing effort to address accessibility in a wide range of contexts.

This research focuses specifically on the accessibility of digital events, examining key factors such as the creation of digital content that is fully perceivable and understandable, as well as the platforms used to host these events. The importance of accessibility in digital events has become particularly evident in recent years, especially during the Covid-19 pandemic. Restrictions related to the public health emergency accelerated the shift from in-person to online formats for conferences, lectures, seminars, and other events. As a result, videoconferencing platforms such as Zoom, Google Meet, and Microsoft Teams quickly became essential tools for participation and collaboration.

This shift to digital formats has also led to the emergence of new barriers for people with disabilities, who are often excluded from digital experiences that were not designed with accessibility in mind. The need to ensure that digital content, such as PowerPoint presentations and video streams, is both perceivable and usable by all is now recognised not only as a fundamental right, but also as a legal obligation, particularly with the upcoming enforcement of the European Digital Act[5].

While many platforms for webinars have introduced accessibility features, significant challenges remain. Common issues include insufficient support for screen readers, poor keyboard navigation, lack of alternative text for images, and inadequate colour contrast, all of which can hinder participation for users with disabilities. This research examined current accessibility guidelines and best practices, with a particular focus on how they apply to the main components of digital event platforms: video players, presentation viewers, and chat component. By referencing the Web Content Accessibility Guidelines (WCAG) 2.2 and other resources, we identified a set of criteria that digital platform components must meet to ensure accessibility. We also recommended best practices to help improve the overall accessibility of these platforms.

Starting on these findings, we developed a tool to convert PowerPoint presentations into accessible HTML pages. This solution aims to overcome the limitations of traditional screen sharing, which does not allow users of assistive technologies to independently access shared content.

Finally, we tested our tool using a set of slides and benchmarked the supported features. The HTML pages were submitted to accessibility checks to verify compliance with WCAG and their usability

ACM ISBN 979-8-4007-2089-5/25/09
https://doi.org/10.1145/3748699.3749815

with screen-readers and keyboard navigation. We also calculate the amount of bandwidth that can be saved with the proposed solution.

The paper is organised as follows: Section 2 describes the related work and background. Accessibility issues and improvements in guidelines for in digital events are discussed, respectively, in Section 3 and Section 4. Section 5 presents a tool for converting PowerPoint presentations into HTML pages and Section 6 reports tests and the obtained results. We conclude in Section 7.

2 Background and Related Work

The Web Content Accessibility Guidelines (WCAG)[9], developed by the World Wide Web Consortium (W3C), establish the foundational principles of web accessibility. These guidelines are structured around four key principles, i.e., Perceivable, Operable, Understandable, and Robust, and define a comprehensive set of criteria to ensure that digital content is accessible to users of all abilities. In addition to these guidelines, the Web Accessibility Initiative (WAI) offers additional resources which outline the best practices for media accessibility, particularly for audio and video content. Achieving real accessibility depends not only on the design of digital services but also on the accessibility of the content itself [16]. To ensure that content is usable by all individuals, several key practices should be implemented: providing accurate captions for all video content; including audio descriptions for capturing visual elements that are essential to understanding; and offering transcripts for both audio and video materials [8] [10] [25].

Despite these initiatives, a recent report highlights that most websites nowadays still fail to meet basic accessibility requirements, indicating that the path toward a fully inclusive web remains challenging [26]. For this reason, multiple institutions have pushed the conversion of these guidelines into legislation. In the US, Section 508 of the US Rehabilitation Act requires federal agencies to ensure that their Information and Communication Technology (ICT), including websites, documents, software, and hardware, is accessible to people with disabilities [1]. Similarly, the European Accessibility Act, effective from June 2025, requires compliance with European Standard EN 301 549, i.e., with WCAG 2.1 Level AA criteria. Its primary goal is to harmonise accessibility regulations across all European countries [5].

Within this legislative context, webinars and digital events must adhere strictly to these accessibility standards. However, recent studies showed that the major video conferencing platforms, such as Zoom, Microsoft Teams, Google Meet, and Skype, still have persistent accessible barriers. In 2024, Hersh et al conducted a comparative study of the accessibility and usability of these platforms with 81 people with disabilities. The study reported that, despite some platforms being preferred to others in terms of usability, there was often an inadequate or inaccurate caption, insufficient support for screen reader navigation, and problematic keyboard accessibility across all platforms [15].

Additionally, a significant challenge for digital events concerns screen-sharing functionalities. Screen reader users frequently require external support to effectively present their material during screen-sharing sessions. Moreover, screen reader users have encountered considerable issues in accessing and comprehending visually shared content. The lack of textual descriptions or accessible alternatives leaves the accessibility of the shared information dependent solely on the speaker's verbal communication, often resulting in inaccessible content. Danyang et al [7] proposed two prototypes to provide adequate ways to access screen-shared slide content during presentations through screen readers, giving users control over what, when, and how slide elements are read out and introducing spatial audio separation between the presenter's voice and screen reader audio. The participants in the user study highlighted the importance of concurrent exploration and of understanding information using alternative means.

Presenters commonly use slides as visual aids for informative talks. When presenters fail to verbally describe the content on their slides, blind and visually impaired audience members lose access to necessary content, making the presentation difficult to follow. An analysis of 90 presentation videos conducted by Peng et al [18] revealed that 72% of 610 visual elements (e.g., images, text) were insufficiently described. They proposed Presentation A11y to help presenters create accessible presentations. The system analyses visual elements on the slide and the transcript of the verbal presentation to provide element-level feedback on what visual content needs to be further described or even removed, thus helping the presenters to improve the accessibility of informational content for all. On the other hand, I-Scratch [13] is a multimodal system which allows people with visual impairments to independently create, explore, and edit PowerPoint slides. It was designed through iterative participatory sessions involving a blind user. It integrates a graphical tactile display with auditory guidance for multimodal feedback and leverages AI technologies for visual assistance in image generation and content interpretation.

Other studies have explored the use of web browsers as presentation viewers, with the aim of demonstrating the versatility of a component that is now available on everyday devices [3, 20, 23, 27]. In 2015, Ciman et al. developed a system that transforms PowerPoint slides into interactive HTML5 presentations [3]. They aimed to increase audience engagement through real-time polls and feedback during webinars and digital events. However, the system does not focus on accessibility features, such as screen reader compatibility or semantic navigation.

Similarly, in 2014, Kucukosman and Yilmaz designed the *Office Tagger*, a tool to generate *.pptx* files starting from XML inputs and therefore automate presentation creation. The emphasis of this research was on generating documentation rather than on ensuring front-end accessibility according to users' needs [27]. Lastly, Vilimek's work on PPTX-to-HTML conversion showcases a web application that enables PowerPoint viewing in browsers and supports various features, including text, images, and multimedia. The main focus of this research was rendering PPTX files with precision; it does not address WCAG compliance or assistive technology support [23].

Our goal is to provide a comprehensive synthesis of relevant accessibility guidelines for digital events. Moreover, the study presents a tool specifically developed to generate accessible HTML code from speakers' PowerPoint presentations. This tool aims to mitigate the limitations of features such as screen sharing and make the shared content available to everyone in real time. Despite the existence of similar converters, the primary goal of our tool is the integration of accessibility into the conversion process, a feature that sets it apart from previous studies and aligns with current laws.

3 Accessibility Issues in the Digital Content

Since our tool converts PowerPoint presentations into web pages, we begin our investigation with an overview of the most common accessibility issues affecting modern websites and digital services, and explain how these issues manifest within digital event platforms. Building on this analysis, we identify and examine a set of key components that are essential to accessibility in online events. These components are assessed against the WCAG 2.2 to develop a framework to evaluate their compliance and inform the developers of accessible alternatives.

The increasing reliance on digital platforms for webinars and online events has highlighted persistent accessibility issues that affect a wide range of users, particularly those with disabilities. According to the 2024 WebAIM Million report, which evaluates the homepages of the top one million websites, 96.3% still fail to meet WCAG compliance, with an average of 56.8 accessibility errors per page. The most common issues include low contrast text (83.9%), missing alternative text (58.1%), and empty links or buttons (50.1%) — all of which represent critical accessibility failures and poor design practices [26].

To raise awareness and promote inclusive design, several institutions have published practical guidance on avoiding common accessibility mistakes. In recent years, organisations such as the UK Government, Queen's University (Canada), and the University of Glasgow have identified a recurring set of issues that undermine web accessibility [4, 17, 19]. These include:

- **Missing alternative text**: prevents screen reader users from accessing the meaning of visual content;
- **Lack of skip navigation**: forces keyboard users to tab through every element without the ability to jump to main content;
- **Inaccessible pop-ups**: trap focus or prevent assistive technologies from accessing their contents;
- **Missing form labels**: make it impossible for screen reader users to identify form fields and input requirements;
- **Insufficient keyboard support**: renders applications unusable for users who cannot operate a mouse;
- **Lack of user testing**: results in interfaces that overlook real-world accessibility needs.

Although most of these guidelines target traditional websites, their relevance extends to digital event platforms, which often feature embedded, dynamic components such as chat interfaces, media players, and presentation viewers. These elements must adhere to accessibility standards to ensure full participation for all users.

Unlike conventional websites, digital event platforms frequently rely on proprietary software and complex interfaces, making their accessibility more difficult to evaluate. Moreover, documentation on the accessibility of such platforms is often scarce. Given the growing importance of online events, we adopt a focused analytical approach to assess accessibility by identifying and evaluating the essential components of these platforms according to WCAG 2.2 Level AA criteria [9], supported by best practice documentation from the WAI [10, 11].

Through this process, we identified three core interactive components that are consistently present across major platforms such as Zoom, Microsoft Teams, and Google Meet:

- **Video Player** — used for broadcasting live or recorded media content;
- **Presentation Viewer** — used for sharing visual materials such as slides, diagrams, and charts;
- **Live Chat** — used for real-time interaction between participants and speakers.

Each of these components was analysed individually against relevant WCAG 2.2 success criteria to develop a tailored framework to support the design and evaluation of accessible digital events.

4 Proposed Solution for Accessible Digital Events

In this section, we report the analysis conducted to evaluate the accessibility of digital event platforms, with a focus on webinars and online events. Our analysis is divided according to each element of interest of a digital platform.

4.1 Video Player

The Video Player is a main component for digital events as it allows the playback of live or pre-recorded content. For this reason, its interface and functionality must be designed to offer a user experience suitable for everyone. According to WCAG 2.2, the best practices which must be implemented are:

- Label in Name: the video player has to show its current state using descriptive labels (e.g., playing, paused, stopped). Screen reader users benefit from this feature (WCAG 2.5.3, Level A).
- Keyboard Navigation and Focus Visibility: player controls must be accessible through keyboard navigation. The focus on each element should also be clearly perceivable. These features facilitate the use of the player for keyboard users and assistive technology users (WCAG 2.1.1 and 2.4.3, Level A; WCAG 2.4.7, Level AA).
- Play and Pause Controls: Users should be able to start and stop video playback. (WCAG 2.2.2, Level A).
- Perceivable Audio and Video Media Content: ensure the media content reproduced can be accessible to everyone by integrating features, such as closed captions for pre-recorded and live videos (WCAG 1.2.1, Level A; 1.2.4, Level AA), transcripts for audio-only content (WCAG 1.2.1 and 1.2.9), and audio descriptions for essential visual content in pre-recorded videos (WCAG 1.2.5, Level AA).
- Avoid Rapid Flashing: ensure that any content does not contain more than three flashes within a one-second (WCAG 2.3.1, Level A).
- Contrast of Captions and Transcripts: ensure a sufficient colour contrast (minimum ratio of 4.5:1) between captions or transcripts and their background (WCAG 1.4.3, Level AA).

Adherence to these criteria is imperative for ensuring the accessibility and usability of the video player and its content for all participants.

4.2 Chat

The chat function is a key component of digital events, enabling participants to engage with speakers, ask questions, and report technical issues in real time. This feature is provided by all current platforms for online conferences such as Zoom, Microsoft Teams, and Google Meet. Any chat must respect the following criteria:

- Alternative Text: provide descriptive alternative texts for icons and images (WCAG 1.1.1, Level A).
- Contrast and Text Resizing: ensure sufficient text contrast (minimum 4.5:1, WCAG 1.4.3, Level AA) and support text resizing without content loss (WCAG 1.4.4, Level AA).
- Avoid Colour-Only Indications: do not rely only on colours to convey information, forgetting users with colour vision deficiencies (WCAG 1.4.1, Level A).
- Keyboard Navigation: all chat functionalities must be fully operable via keyboard (WCAG 2.1.1, Level A; 2.1.2, Level A).
- Clear Headings and Simple Language: use of descriptive headings and simple language for messages to users (WCAG 2.4.6, Level AA; 3.1.5, Level AAA).
- Status Updates: allow also users who use assistive technologies to receive updates on message delivery or new incoming messages using ARIA elements, i.e., `aria-live` (WCAG 4.1.3, Level AA).

Furthermore, the chat may be presented through either a static or a collapsible question box, depending on the platform's interface design. The choice of design can influence accessibility. For instance, a collapsible question box can assist users with motor or low vision by reducing on-screen clutter and simplifying the interface. This can enhance focus and ease of navigation, particularly when assistive technologies are used.

4.3 Presentations

Organisers and speakers commonly use presentations to deliver a speech to the audience, enriched by visual content, i.e., images, charts, and diagrams. Each slide represents a specific topic or section of an argument, allowing both speakers and viewers to follow the flow of the speech. For this reason, slides should be concise, focusing on key points and avoiding superfluous details, in order to present viewers with the essential information. Additionally, tools such as Microsoft PowerPoint and Google Slides have introduced new functions to check basic accessibility on the slides, aligning with WCAGs criteria.

Presentation must comply with the following criteria:

- Alternative Text: provide alternative texts for non-decorative images, diagrams, and graphs, enabling screen reader access (WCAG 1.1.1, Level A).
- Text Contrast and Resizing: keep a sufficient text-to-background contrast (minimum 4.5:1 contrast ratio; WCAG 1.4.3, Level AA), and ensure text can be resized without loss of content or functionality (WCAG 1.4.4, Level AA).
- Avoid Colour-Only Communication: do not rely only on colour to convey essential information, to help users with colour vision impairments (WCAG 1.4.1, Level A).
- Keyboard Navigation and Logical Structure: ensure slides are fully navigable via keyboard. Elements should have a logical and intuitive focus order consistent with visual order (WCAG 2.1.1, Level A; 2.4.3, Level A).
- Descriptive Headings: use unique and descriptive titles and labels for slides, to facilitate navigation and comprehension (WCAG 2.4.6, Level AA).
- Simple language and clear instructions: use clear, simple language, appropriate for the audience (WCAG 3.1.5, Level AAA). Provide explicit instructions and avoid content that flashes or moves too quickly, as this can be a problem for users with photosensitive epilepsy (WCAG 2.3.1, Level A).

Lastly, it is essential to choose fonts that enhance readability, such as sans-serif fonts (e.g., Arial, Verdana, Helvetica) or accessibility-focused fonts like OpenDyslexic. Recommended font sizes are at least 16 pixels for body text, in accordance with the default font size of modern browsers. Line spacing should be set to 1.5 times the font size, with word spacing of at least 0.16 times the font size, and letter spacing of at least 0.12 times the font size. The use of this style setting supports readability for users with visual or cognitive impairments.

4.4 Best Practices for Digital Events

Adhering to accessibility criteria not only ensures compliance with current legislation, such as the European Accessibility Act [5], but also significantly improves the inclusiveness of digital events. However, achieving true accessibility requires attention beyond platform design and content formatting—it also involves how organisers, speakers, and participants approach the event. For this reason, the Web Accessibility Initiative reports a set of best practices to support inclusive digital events for all participants, including those with visual, auditory, motor, or cognitive disabilities [11].

For speakers, it is essential to describe all visual content (e.g., images, charts, diagrams) clearly and explicitly, ensuring that participants with visual impairments can access the full context of the presentation. Additionally, speakers should ensure their slides and materials follow the accessibility principles detailed in the previous sections and provide captions or transcripts for multimedia content. Sharing presentation materials in advance, in accessible formats, further enhances participant engagement.

Organisers hold the responsibility of selecting platforms that support assistive features such as real-time captioning and interpreters. During the event, it is crucial to maintain clear speech at a steady pace, allow extra time for responses, use high-quality microphones, and minimise distractions.

Despite these efforts, some features, such as live screen sharing, can unintentionally compromise accessibility. E.g., screen readers cannot access the visual content displayed in a shared screen, preventing users with visual impairments from independently navigating the material. To address this gap, we developed a dedicated tool that converts Microsoft PowerPoint presentation files into accessible HTML pages, enabling participants to interact directly with the content in their browsers. Moreover, shared screens are transmitted as video streams that require more bandwidth than simple HTML pages.

Section 5 describes the design and implementation of this converter, highlighting how it enhances access to presentation materials and ensures that users relying on assistive technologies do not lose any essential information.

5 Converter

We developed a new tool designed to convert *.pptx* files into accessible HTML pages. The tool is developed entirely in JavaScript and runs on the Node.js runtime environment.

The supported input format is Microsoft PowerPoint (*.pptx*), which conforms to the ECMA-376 Open XML standard [6] . This format encapsulates presentation data within a ZIP archive containing a structured collection of XML files. These files define the presentation's content, layout, styles, embedded media, and metadata.

By parsing XML files, the tool systematically extracts both the logical structure and visual elements of the presentation. This information is then processed and stored in memory before being used to construct a corresponding HTML page. The output consists of a clean, readable, and accessible HTML file, accompanied by CSS stylesheets to accurately replicate the visual formatting of original slides.

Each PowerPoint element is translated into a semantic HTML tag, ensuring that the resulting page is both visually faithful to the source but also compliant with web standards and WCAG.

Our converter currently supports the following PowerPoint features:

- **Text Boxes**: text boxes are extracted from the `<p:sp>` elements (which represent shapes) and the `<a:t>` tags that contain the actual text. These are converted into HTML `<p>` elements. The style information—such as font, colour, alignment, and spacing—is taken from child elements like `<a:rPr>` (run properties) and `<a:pPr>` (paragraph properties).
- **Shapes**: detected by the `<p:sp>` element, with geometric definitions specified under `<a:prstGeom>` (preset geometry). The system recreates these shapes in HTML using SVG tags to preserve their vector properties.
- **Pictures**: identified within the `<p:pic>` element, where the image reference is found under the `<a:blip>` tag with the `r:embed` attribute. Additional information such as position, size, and rotation is then retrieved from the `<a:xfrm>` tag.
- **Background Images**: managed through the `<p:bg>` element, which can include a background image reference using `<a:blipFill>` and the associated `<a:blip>` tag.
- **Tables**: Parsed from the `<p:graphicFrame>` element when the embedded content is a table, with individual cells represented by `<a:tc>` (table cell) and their text content nested inside `<a:t>` tags.

5.1 System Architecture

The system architecture is organised into the following core modules:

- **Converter-engine**: is the module responsible for handling the input files, executing the analysis of *.pptx* elements, and producing the HTML output. It coordinates all submodules of the converter;
- **XML-parser**: the main component of the converter. It is responsible for reading and interpreting the XML files contained within the *.pptx* file. It extracts the elements according to their types and maps them into internal data representations;
- **Slide-creator**: the module responsible for converting the elements of the slides into data which will be placed on the generated HTML files;
- **Element-placer**: it places all the elements on the corresponding HTML pages. It writes a set of CSS rules to associate the style settings of these elements, according to their properties.

These modules read, therefore, every element contained in each slide, parse the information included in the XML files, and translate it into HTML.

For example, when the parser encounters an image element, the module identifies the image source via the `r:embed` attribute inside the `a:blip` tag, i.e., the id `rId2`. Then, it finds the actual image file by resolving this id inside the corresponding slide relationships file (.rels). Lastly, it collects the properties about position, size, and rotatios by reading the `<a:xfrm>` tag. The tag contains the attribute `rotations`, for this property, and it has as children the tag `<a:off>`, which specify the `x` and `y` offset. Lastly, it captures the width and height information by reading the `<a:ext>` tags.

The data are stored in English Metric Units. A EMU is defined as 1/360.000 of a centimetre and thus there are 914.400 EMUs per inch, and 12.700 EMUs per point. The collected data is then transformed into CSS properties and embedded within semantic HTML elements. Each converted object is wrapped inside a uniquely identified `<div>` container, ensuring precise styling and clear object type annotation.

Listing 1 shows a piece of XML code of a slide containing a picture with the alternative text.

```xml
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="5" name="Picture 4"
    descr="Two walnut trees on a light green meadow with
     a blue sky and a few clouds in the background.">
      ...
    </p:cNvPr>
    </p:nvPicPr>
    <p:blipFill>
        <a:blip r:embed="rId2">
            ...
        </a:blip>
        ...
    </p:blipFill>
    <p:spPr>
        <a:xfrm>
           <a:off x="3156155" y="1691150"/>
           <a:ext cx="5577329" cy="4253660"/>
        </a:xfrm>
        <a:prstGeom prst="rect">
            <a:avLst/>
        </a:prstGeom>
    </p:spPr>
</p:pic>
```

Listing 1: Example of picture in XML file.

5.2 Accessibility Integration

Accessibility integration relies on two factors. Firstly, it depends on accessibility compliance, which relies on the original *.pptx* files. Secondly, depends on the structure of the HTML code generated by the parser.

Nowadays, several commercial products, such as Aspose[2] and iSpring[12], translate *.pptx* files into HTML pages with an extreme precision of the rendering of the aspect of the original files. But the resulting output is an HTML code cluttered and difficult to interpret, not accessible or compatible with screen readers. For example, iSpring, for instance, includes an accessibility feature that extracts textual content and images from the slides and presents them within a carousel interface to comply with WCAG guidelines. While this approach enhances accessibility, it significantly alters the original structure and layout of the presentation. In contrast, our tool is designed to produce HTML presentations that are both WCAG 2.2 compliant and faithful to the original slide structure. By preserving the visual layout and semantic structure defined in the source file, our converter ensures accessibility without compromising the integrity of the original presentation. A main focus was the usability for screen reader users and keyboard navigation. Each slide preserves the logical reading order defined in the original PowerPoint file, which is essential for ensuring that screen readers present content in a coherent and meaningful sequence. This reading order is derived from the element hierarchy and properties embedded in the *.pptx* file.

To support navigation, the generated HTML pages include two buttons for moving to the previous or the next slide. These buttons are fully operable via keyboard and are correctly announced by screen readers. After a new slide is loaded, the focus is systematically set to the first element on the slide, enabling a seamless and autonomous browsing experience for users relying on assistive technologies.

Accessibility-related metadata is extracted directly from the XML files. For example, alternative text for images is retrieved from the `descr` attribute of the `<p:cNvPr>` tag, following the ECMA-376 standard. This attribute provides descriptive text for DrawingML objects, which is essential for screen reader interpretation when the visual content cannot be perceived.

The parser translates the XML content into semantically appropriate HTML elements:

- Titles are rendered using `<h1>` tags,
- Paragraphs with `<p>`,
- Images with `<img>`,
- Shapes with `<svg>`,
- List with `<ul>` and `<li>`.

This semantic structure ensures that the generated HTML is both compliant with accessibility standards and effectively interpretable by assistive technologies, thereby improving the overall usability and navigation experience for all users.

The code reported on Listening 1 is then parsed by our tool as shown in Listening 2.

```
<div id="slide2_img_layer4_id" class="img visible">
    <img id="img_slide2_4"
    src="../img/image2.jpg"
    alt="Two walnut trees on a light   green meadow with
     a blue sky and a few clouds in the background.">
</div>

#slide2_img_layer4_id {
  position: absolute;
  width: 586px;
  height: 447px;
  z-index: 4;
  transform: rotate(0deg);
  overflow: hidden;
  left: 331px;
  top: 178px;
}
```

Listing 2: Example of HTML snippet

As we can see, the converter properly stores information about the image's position, size, and source. The `descr` attribute from the `<p:cNvPr>` tag is mapped directly to the HTML `alt` attribute, ensuring that screen readers can convey meaningful descriptions of the image to visually impaired users. Additionally, all positional and size data are accurately converted from EMU units to pixels, preserving the visual layout and maintaining fidelity with the original slide design.

6 Results

This section reports the outcomes of the evaluation conducted on our tool. The analysis includes both quantitative and qualitative assessments.

The quantitative evaluation focuses on the detection and accurate rendering of presentation features, measuring the percentage of elements correctly identified and reproduced in the HTML output.

The qualitative evaluation examines the overall quality of the generated HTML pages, with particular attention to visual fidelity compared to the original PowerPoint files, as well as compliance with accessibility standards such as WCAG 2.1 Level AA.

6.1 Quantitative Analysis

Our quantitative analysis considers six different presentations, totaling 145 slides. Each presentation was created using Microsoft PowerPoint by a different author, resulting in a diverse set of features and styles. Authors come from various backgrounds, i.e., professors and students from the University, sales managers and project leaders from four different companies. Table 1 summarises the detection and parsing accuracy of presentation elements.

Table 1: Element detection, parsing, and accuracy results

Number of	Images			Shapes			Text boxes			Tables		
slides	Det.	Pars.	%	Det.	Pars.	%	Det.	Pars.	%	Det.	Pars.	%
24	69	69	100%	102	90	88.2%	94	94	100%	1	1	100%
39	78	78	100%	379	332	87.6%	211	211	100%	2	2	100%
15	12	12	100%	43	36	83.7%	57	57	100%	0	0	–
18	46	46	100%	67	47	70.1%	46	46	100%	0	0	–
16	41	41	100%	24	24	100%	46	46	100%	0	0	–
33	168	168	100%	242	222	91.7%	172	172	100%	0	0	–
Total 145	414	414	100%	857	751	87.6%	626	626	100%	3	3	100%

The table reports one row per presentation and groups results by element type - shapes, text boxes, images, and tables. For each type, we report the number of elements detected, parsed, and resulting accuracy. The *detected* column refers to the number of elements identified by our parser based on specific XML tags within the *.pptx* file:

- `p:txBody` for text boxes,
- `a:prstGeom` and `p:cxnSp` for shapes,
- `p:pic` for images,
- `p:graphicElements` for tables.

The *parsed* column indicates the number of elements that were rendered in the final HTML output. This number does not consider placeholders (e.g., "Add a Title") and other layout settings that do not contribute to the slide content.

Data shows that the tool reaches 100% parsing accuracy for images, tables, and text boxes across all presentations. In contrast, shapes got a lower success rate (87.6% overall), due to the partial support of shape types implemented in our parser. In particular, at the actual stage of development, our tool supports a set of basic geometric shapes which include: rect, line, parallelogram, arrow, straightConnector, ellipse, frame, triangle, roundRect, and bentConnector. Any shape defined by using a different a:prstGeom value is therefore not rendered in the final HTML output. For example, the first presentation analysed in Table 1 includes decorative stars, which fall outside our supported list.

Despite the current limitations in shape support, the tool demonstrates high reliability, achieving near-complete rendering of presentation content. Future work will focus on extending support to additional shape types to further increase accuracy and coverage.

6.2 Qualitative Analysis

Our qualitative analysis focused on evaluating the accessibility of the converted slides using usability tests, which involved real users. An accessibility expert evaluated the resulting HTML pages using screen reader technology and keyboard navigation, achieving encouraging results. Screen reader tests were conducted using Non-Visual Desktop Access (NVDA), version `2024.4.2`, a free and open-source screen reader widely referenced in accessibility evaluation contexts, including WebAIM [24]. A key aspect of the evaluation involved verifying the logical focus order of interactive elements when navigating the slides via assistive technologies.

Figure 1 shows the focus sequence generated by NVDA on an HTML slide. It is important to note, firstly, that the focus order mirrors the reading order defined in the original PowerPoint (*.pptx*) file. In this example, the screen reader first announces the slide's title, which corresponds to the first element in the structure of the original presentation and is translated into a `h1` tag. It then moves on to a paragraph, which is parsed and rendered as a `p` tag in the HTML output. Next, the screen reader shifts focus to two images, both of which include alternative text. NVDA properly reads the `alt` text for each image, ensuring the transmission of information contained in visual elements through audio. This confirms that the semantic structure and metadata required for screen reader interpretation were properly preserved during the conversion process. Finally, the focus advances to the two interactive buttons *Previous Slide* and *Next Slide*. Although these buttons are not part of the original PowerPoint file, they were intentionally added during conversion to allow users to have control over the current slides to view. Further tests were carried out using automated accessibility testing

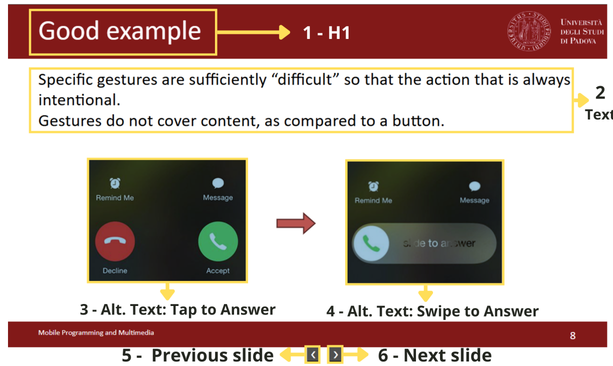

Figure 1: Example of focus order.

tools, including Silktide Accessibility Checker and TotalValidator. Silktide is a browser extension for Google Chrome that performs comprehensive accessibility checks based on WCAG compliance [14]. It provides a set of checks that includes the presence of alternative text for images, colour contrast ratios, and compatibility with screen readers. In contrast, TotalValidator is a standalone tool that performs in-depth HTML and accessibility validation. It scans not only individual pages but also their associated internal links and resources [22].

The results of these tests confirmed that our parser does not introduce new accessibility errors. However, any accessibility issues identified in the original PowerPoint presentations, such as missing alternative text, insufficient colour contrast, or wrong focus order, are also present in the resulting HTML files. In summary, the tool preserves the structure and content of the source files without compromising accessibility.

Furthermore, the tests confirmed that the generated HTML files conform to HTML5 syntax and WCAG 2.2 Level AA standards. The converted pages have a valid document structure with correctly nested HTML elements, as well as a semantically appropriate use of tags and CSS styles.

7 Conclusion

Ensuring accessibility in digital events is not solely a compliance with legislation, but it is a fundamental aspect of achieving inclusion. This paper explores the accessibility challenges for online platforms by integrating current guidelines such as WCAG 2.2 and WAI best practices with a comprehensive analysis of their application to digital event components. We started by identifying the most common accessibility issues found on the web and examined how they affect key features of digital events, like video players, presentation viewers, and live chat. Based on this analysis, we created a set of guidelines and best practices to help design platforms for accessible digital events.

Based on our findings, we developed a tool that converts PowerPoint presentations into accessible HTML pages. The goal was to ensure that content shared during presentations is both accessible and perceivable. Traditional screen sharing, by contrast, does not allow users who rely on assistive technologies to independently access the shared content, making them dependent solely on the speaker's verbal descriptions, which are often insufficient [18]. Unlike screen sharing, our tool preserves the semantic structure of slides, supports screen reader navigation, and enables full keyboard interaction. The resulting HTML is clean, standards-compliant, and meets WCAG 2.2 Level AA requirements.

We conducted a series of tests to assess the performance of our converter. The evaluation showed both high parsing accuracy and string accessibility performance. The converter faithfully replicates the original structure of the presentation without introducing any additional accessibility issues. All problems identified in the converted files address the issues already present in the input source files.

Compatibility with NVDA screen reader, SilkTide features, and TotalValidator tests confirmed the value of our product. The integration of HTML presentation into webinars can be a game-changer in the realm of digital events, making the content accessible to everyone.

Moreover, the use of HTML presentations has the potential to result in a big reduction of bandwidth in such scenarios when video is not essential. This is achieved by integrating the slides directly onto the platform page, thus avoiding the use of screen-sharing features and the transmission of heavy video streams: e.g., a video lesson of one hour and 18 minutes requires transmitting a video of 224,4 MB (without audio). The corresponding HTML file obtained with our tools requires only 17,4 MB, thus saving more than 92% of the bandwidth.

Overall, this research offers both practical guidelines and a working solution to improve accessibility in digital events. Future improvements will focus on extending the features supported by our parser, including complex shapes and interactive features like transitions and animations.

Acknowledgments

This research was supported SMACT and PrimoRound with the project "Roadmap to ADE (Accessible Digital Events) - R2ADE". Responsibility for the content resides with the authors.

References

[1] U.S. General Services Administration. 2024. Section 508: Laws and Policies. https://www.section508.gov/manage/laws-and-policies/. Accessed: 2024-05-22.

[2] Aspose. 2024. Free Online PPTX to HTML Converter. https://products.aspose.app/slides/conversion/pptx-to-html. Accessed: 2024-05-24.

[3] Matteo Ciman, Yari Formaggio, Ombretta Gaggi, and Marco Regazzo. 2015. May SmartPhones Help to Maintain Audience Attention During Presentations? , 55-63 pages. https://doi.org/10.5220/0005441100550063

[4] Kelly Clarkson. 2025. 10 Digital Accessibility Mistakes to Avoid. https://accessibility.blog.gov.uk/2025/02/04/10-digital-accessibility-mistakes-to-avoid/. Senior Accessibility Specialist, Government Digital Services.

[5] European Commission. 2024. European Accessibility Act. https://commission.europa.eu/strategy-and-policy/policies/justice-and-fundamental-rights/disability/union-equality-strategy-rights-persons-disabilities-2021-2030/european-accessibility-act_en. Accessed: 2024-05-22.

[6] Ecma International. 2021. ECMA-376: Office Open XML File Formats. https://ecma-international.org/publications-and-standards/standards/ecma-376/. Standard ECMA-376, 5th edition.

[7] Danyang Fan, Sasa Junuzovic, John Tang, and Thomas Jaeger. 2023. Improving the Accessibility of Screen-Shared Presentations by Enabling Concurrent Exploration. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA) *(ASSETS '23)*. Association for Computing Machinery, New York, NY, USA, Article 44, 16 pages. https://doi.org/10.1145/3597638.3608411

[8] W3C User Agent Accessibility Guidelines Working Group. 2015. *User Agent Accessibility Guidelines (UAAG) 2.0*. Technical Report. World Wide Web Consortium (W3C). https://www.w3.org/TR/UAAG20/ W3C Recommendation, Accessed: 2024-05-22.

[9] World Wide Web Consortium Web Accessibility Initiative Group. 2008. Web Content Accessibility Guidelines (WCAG) 2.2. https://www.w3.org/TR/WCAG22/

[10] W3C Web Accessibility Initiative. 2023. Planning Audio and Video Media Accessibility. https://www.w3.org/WAI/media/av/planning/. Accessed: 2024-05-22.

[11] W3C Web Accessibility Initiative. 2024. How to Make Your Presentations Accessible to All. https://www.w3.org/WAI/teach-advocate/accessible-presentations/. Accessed: 2024-05-23.

[12] iSpring Solutions. 2024. iSpring Converter Pro. https://www.ispringsolutions.com/ispring-converter. Accessed: 2024-05-24.

[13] Gyeongdeok Kim, Chungman Lim, and Gunhyuk Park. 2025. I-Scratch: Independent Slide Creation With Auditory Comment and Haptic Interface for the Blind and Visually Impaired. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1161, 23 pages. https://doi.org/10.1145/3706598.3713553

[14] Silktide Ltd. 2024. Silktide Toolbar: Accessibility Checker for Chrome. https://silktide.com/toolbar/. Browser extension for WCAG accessibility evaluation.

[15] Marina Buzzi Marion Hersh, Barbara Leporini. 2024. A comparative study of disabled people's experiences with the video conferencing tools Zoom, MS Teams, Google Meet and Skype, Behaviour and Information Technology. https://doi.org/10.1080/0144929X.2023.2286533. Accessed: 2024-05-22.

[16] Allard Oelen and Sören Auer. 2019. Content Authoring with Markdown for Visually Impaired and Blind Users. In *2019 IEEE International Symposium on Multimedia (ISM)*. 285–290. https://doi.org/10.1109/ISM46123.2019.00064

[17] University of Glasgow. 2024. Digital Accessibility. https://www.gla.ac.uk/myglasgow/digitalaccessibility/. University of Glasgow Digital Accessibility Hub.

[18] Yi-Hao Peng, JiWoong Jang, Jeffrey P Bigham, and Amy Pavel. 2021. Say It All: Feedback for Improving Non-Visual Presentation Accessibility. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 276, 12 pages. https://doi.org/10.1145/3411764.3445572

[19] Queen's University. 2024. Top Website Accessibility Errors. https://www.queensu.ca/accessibility/tutorials/website-accessibility/top-website-accessibility-errors. Accessibility Hub, Queen's University.

[20] Adam Strantz. 2021. Using Web Standards to Design Accessible Data Visualizations in Professional Communication. *IEEE Transactions on Professional Communication* 64, 3 (2021), 288–301. https://doi.org/10.1109/TPC.2021.3091784

[21] United Nations. 2006. Convention on the Rights of Persons with Disabilities.

[22] Total Validator. 2024. Total Validator: HTML and Accessibility Validator. https://www.totalvalidator.com/. Tool for validating HTML, accessibility, and broken links.

[23] Hynek Vilímek. 2016. PPTX to HTML Conversion. In *Proceedings of Excel FIT 2016*. Faculty of Information Technology, Brno University of Technology. https://excel.fit.vutbr.cz/submissions/2016/038/38.pdf Accessed: 2024-05-23.

[24] WebAIM. 2020. Using NVDA to Evaluate Web Accessibility. https://webaim.org/articles/nvda/. Last updated July 2020.

[25] WebAIM. 2024. Captions, Transcripts, and Audio Descriptions. https://webaim.org/techniques/captions/. Accessed: 2024-05-22.

[26] WebAIM. 2024. The WebAIM Million: An annual accessibility analysis of the top 1,000,000 home pages. https://webaim.org/projects/million/. Accessed: 2024-05-22.

[27] Fatma YILMAZ and Fatma Nur KUCUKOSMAN. 2014. OFFICE TAGGER : A PowerPoint Generating Web Application. , 41 pages.