2022

Exploring Accessibility Features and Plug-ins for Digital Prototyping Tools

Urvashi Kokate
Rochester Institute of Technology

Kristen Shinohara
Rochester Institute of Technology

Garreth W. Tigwell
Rochester Institute of Technology

Recommended Citation

Urvashi Kokate, Kristen Shinohara, and Garreth W. Tigwell. 2022. Exploring Accessibility Features and Plug-ins for Digital Prototyping Tools. In The 24th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '22), October 23–26, 2022, Athens, Greece. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3517428.3550391

Exploring Accessibility Features and Plug-ins for Digital Prototyping Tools

Urvashi Kokate
uk4890@rit.edu
School of Information
Rochester Institute of Technology
Rochester, NY, USA

Kristen Shinohara
kristen.shinohara@rit.edu
School of Information
Rochester Institute of Technology
Rochester, NY, USA

Garreth W. Tigwell
garreth.w.tigwell@rit.edu
School of Information
Rochester Institute of Technology
Rochester, NY, USA

ABSTRACT

Many digital systems are found to be inaccessible and a large part of the issue is that accessibility is not considered early enough in the design process. Digital prototyping tools are a powerful resource for designers to quickly explore both low and high fidelity design mockups during initial stages of product design and development. We evaluated 10 popular prototyping tools to understand their built-in and third-party accessibility features. We found that accessible design support is largely from third-party plug-ins rather than prototyping tools' built-in features, and the availability of accessibility support varies from tool to tool. There is potential to improve accessible design by increasing the potential for accessibility to be consider earlier in the design process.

CCS CONCEPTS

• **Human-centered computing** → **Accessibility**.

ACM Reference Format:
Urvashi Kokate, Kristen Shinohara, and Garreth W. Tigwell. 2022. Exploring Accessibility Features and Plug-ins for Digital Prototyping Tools. In *The 24th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '22), October 23–26, 2022, Athens, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3517428.3550391

1 INTRODUCTION AND RELATED WORK

Accessible design should not be an afterthought in the design process [4]. When accessibility issues are checked later in the design process, it requires more time, money, and effort in redesigning and reprogramming the product [9, 13, 23, 24]. Therefore, it is beneficial to move accessible design as early as possible in the design process [16].

Accessibility tools are an essential part of the design process to support people meeting accessible design standards set by the Web Content Accessibility Guidelines (WCAG) [12], and mobile app design guidelines also incorporate WCAG criteria [2, 7, 17] due to an increasing need for mobile accessibility [1]. Designers and developers have access to a multitude of accessibility tools [5, 10]. However, they generally fall into two categories: those to be used during design and development, and those that run evaluations after development. Automated accessibility checking generally results in missing some issues compared to manual assessment [14], and there is also a lack of consistency among accessibility evaluation tools [6, 18]. However, there is potential with accessibility tools that support people in meeting WCAG (e.g., color contrast checkers), especially if they are designed to be used earlier in the design process (e.g., the Accessible Colour Evaluator, which was developed through a User-Centered Design process [22]).

Prototyping is a design method used early in the design process to lay down ideas and also support quicker evaluations [11, 20]. There are many approaches to facilitating prototyping practice ranging from working offline (e.g., pen and paper) to using low-cost, readily available software (e.g. Microsoft PowerPoint) to high-end professional software (e.g., Adobe XD). We anticipated that prototyping tools are likely not providing enough feature support for users since prior work already found prototyping tools themselves are not made to be accessible [15]. Some prototyping software also supports the use of plug-ins, and there is potential with this approach to offer accessibility plug-ins (e.g., Adee [8]).

There are many scenarios where prototyping tools could support accessible design. For example, notifying the user when color pairs are inaccessible due to not meeting WCAG minimum contrast criteria, warning when button target sizes may be too small for certain devices, recommending accessible fonts styles and font sizes.

We formally evaluated 10 popular design tools that can be used to support digital prototyping to understand what accessibility support they provide. We found that only a few tools have built-in features to support accessible design and most of the tools rely on third party plug-ins that provide features to check for accessibility. The majority focused on color contrast checking and offered problematic color blind simulations. We recommend that companies and the HCI community focus more on providing a range of accessibility check features to support accessible design during on of the earliest points of the design process.

2 EVALUATION OF DESIGN PROTOTYPING TOOLS

Selection of tools. We selected 10 UI design tools from the 2020 design tools survey conducted by UX tools [19]. Our selection included: Adobe Illustrator, Adobe Photoshop, Adobe XD, Affinity Designer, Axure, Figma, Framer, InVision Studio, Sketch, and UXPin.

Evaluation method. Our evaluation process had three steps.

Step 1: Find and document any accessibility-related support reported in the prototyping tool's documentation and/or website. We first browsed through the website and then did a keyword search to look up for specific sections of the page. We selected keywords related to visual design, accessibility, and guidelines since interface

ASSETS '22, October 23–26, 2022, Athens, Greece

prototyping does not often have the ability for audio (see list of keywords in supplementary materials). We first checked the top-level menus and landing page of each tool's website. Then we did a keyword search in the help/support/documentation, and blog section of each website. **Step 2**: Check for any built-in accessible design features in the tools. We checked the top menu, tool bar, and right clicked on design elements to identify any accessibility features the prototyping tools offer. **Step 3**: Check for third-party plug-ins for each tool that supports accessible design. We searched plug-in managers with Step 1's list of keywords. We read the description of each plug-in and noted down its features. We also checked whether it was free, free with paid features, or paid.

2.1 Findings

2.1.1 Searching through each prototyping tool's documentation and website. We found two out of 10 tools (Affinity Designer, Framer) did not have any content related to accessibility. Six prototyping tools (Adobe XD, Axure, Figma, InVision Studio, Sketch, UXPin) featured blogs based on different topics of accessibility such as 'what is accessibility', 'design practices for accessibility' and 'why designers should think about accessibility'. Adobe Illustrator and Adobe Photoshop mention about their inbuilt color blind simulation features in their documentation. UXPin highlights the built-in features of color contrast checker and color blind simulation in the 'features' section of their websites. Adobe XD and Figma also highlight the Stark plug-in which provides different accessibility check features to the users such as color contrast checker, color blind simulator, and focus order.

A majority of the tools provide information on why and how to include accessibility in design, but only half of the tools mention how the tool itself can support them in designing or checking for accessibility. Also, the information provided is limited and hard to find on the website. None of the tools have a section in the top-level menu for accessibility on their website. To look up content, a user needs to search for the content using different keywords. 'Accessibility' and 'contrast' were the keywords that returned informative content about accessibility for seven of the prototyping tools. We based how easy it was to search for information about accessibility on the number of results that were returned and the time taken to find relevant results out of the total returned results. More time was taken to look up content which returned more results. Figma proved more most difficult to find relevant content because searching for 'accessibility' in the help center gave over 200 results out of which only 3 included the word accessibility since most of the results were for the word 'access'. It was easiest to look up for content on UXPin because it returned fewer results and the built-in features were highlighted on the feature page of the website, which was easier to spot.

Table 2 in Supplementary Materials provides a breakdown of accessibility content for each prototyping tool.

2.1.2 Evaluating tool's built-in accessibility features. Affinity Designer and Sketch did not have any built-in accessibility features. We were able to identify five categories that built-in accessibility features would fit. The categories were: **(1)** color contrast checker (found in UXPin); **(2)** adding audio feedback and speech output (found in Adobe XD); **(3)** color blind simulation (found in Adobe Illustrator, Adobe Photoshop, UXPin); **(4)** voice prototyping (found in Adobe XD); and **(5)** adding keyboard navigation (found in Adobe XD, Axure, Figma, Framer, InVision Studio, UXPin).

UX Pin is the only tool that had a built-in color contrast checker that automatically identifies text colors with insufficient contrast with respect to WCAG guidelines. It also had a color blind simulator that allowed users to view their designs in eight different types of color blindness, as well as a prototyping feature to add keyboard shortcut as a trigger to transition across screens. Adobe Illustrator and Adobe Photoshop both provided color blind simulation features, but they only allowed users to view in Protanopia or Deuteranopia. Adobe XD was the only prototyping tool that allows users to add audio feedback and speech output in the design to provide guidance for users who can access the prototype with only sound. Adobe XD also supports voice prototyping, which allows users to design transitions through voice commands. Axure, Figma, Framer, and InVision also had the prototyping feature to add keyboard shortcuts for transition. Table 3 in Supplementary Materials provides an overview.

2.1.3 Evaluating third-party accessibility plug-ins. Affinity Designer, Axure, and UX Pin did not support any third-party plug-ins. We were able to identify three general categories that plug-in accessibility features would fit, and we provide examples for each. Table 4 in Supplementary Materials provides an overview of the plug-in accessibility features category type found for each prototype tool, While Tables 5-11 provide details on each prototyping tool's plug-ins.

Plugins used for **accessibility checking**: *Color contrast checker*: checks contrast between two color layers (foreground text color and background color); *Touch target size checker*: checks touch target size with respect to devices and shows if it violates guidelines; and *Epilepsy checker*: checks if images and animated GIFs in designs are safe for people with photosensitive epilepsy to view.

Plugins used to **enhance accessibility**: *Alt text generator*: adds alt-text for images to share with the developers; *Focus orderer*: adds focus order for keyboard navigation; *Screen reader support*: adds ARIA roles, ARIA properties and tab index in designs; and *Text resizer*: creates legible texts with respect to screen size.

Plugins used to **support understanding impairments**: *Color blind simulation*: tests designs with different color blind simulation; and *Visual Impairment Simulation*: checks elements against different types of visual impairments such as central loss, blind spots, hemianopia, peripheral loss, retinal detachment, ocular albinism.

We list all third party accessibility plug-ins in Table 1 by prototyping tool. Figma offered the most number of plug-ins equal—17 in total. Though Figma had relatively more plug-ins, 11 out of the 17 plug-ins actually only included features to check colors accessibility. Table 4 in Supplementary Materials shows the 17 plug-ins cover seven accessibility feature categories. Most of the plug-ins were free to use. The Stark plug-in is common across the top 3 most popular tools—Adobe XD, Figma, and Sketch. Stark provides color contrast checking, a color blindness simulator, and focus orderer, but there is limited functions within the free version of Stark. The paid features provide smart suggestion of colors to use if the current colors in the design do not adhere to accessibility guidelines. None of the free plug-ins provide color suggestions to the users.

Table 1: List of all third party plug-ins. Details of each tool and plug-in are found in supplementary materials.

Prototyping tools	List of plug-ins
Figma	Able, Zebra, Contrast, A11y - Color Contrast Checker, Color blind, Epilepsy Blocker, Stark, Color Contrast Grid, Cards for Humanity, A11y - Focus Orderer, Adee Comprehensive Accessibility Tool, Contrast, Contrast Grid, HCL Easy, Color Contrast, Low Vision
Sketch	Stark, Adee Comprehensive Accessibility Tool, Cluse, Color Contrast Analyzer, Check Contrast, Color Blindless, Sketch WCAG
Adobe XD	Stark, Colorsinspo, Dopely colors
Adobe Illustrator	Pantone Connect
Adobe Photoshop	Pantone Connect, Check Contrast Ratio
InVision Studio	Contrast
Framer	Color Contrast checker, Color check, Accessibility Tool Kit

Epilepsy blocker is a paid plug-in available only on Figma. It was the only plug-in to allow users to check if images and animated GIFs in designs are safe for people with photosensitive epilepsy. The Adee Comprehensive Accessibility tool plug-in provided the maximum number of features and is available to install in Figma and Sketch only. It was totally free and included features such as alt text generator, touch target size checker, a color blindness simulator and a contrast checker. It also allows users to generate a report for any guideline violations found in a design with respect to touch target size and color contrast. The plug-ins Visual Impairment Simulation and Epilepsy checker were only available for Figma. Screen reader support feature was provided by a plug-in available only on Framer. Color blindness simulators and color contrast checkers were the most popular features offered by plug-ins. Six out of the Seven prototyping tools that support third party plug-ins had color contrast checkers and color blindness simulators.

3 DISCUSSION

We found that although prototyping tools do offer some level of support for accessible design, there is still room for improvement. We found evidence of prototyping tool websites providing information about accessible design practice and info on support their tools offered. However, the information was hard to find since none of the websites had a homepage that linked to an accessibility features support page.

Furthermore, we found the built-in accessibility features in prototyping tools to be limited. The common features were the ability to add keyboard navigation, check color contrast, and run a color blindness simulator. In fact, the majority focused on color contrast checking and offered color blind simulations. We want to acknowledge that color blindness simulations are often limited in accuracy and disability simulations are problematic [3, 21]. It is concerning that so many of the tools and plug-ins focused on color blindness simulations without adequate information on the limitations of using such a feature [21]. Although use of color is a prominent aspect of design, we find that more effort needs to be directed toward creating accessibility features for other accessible design criteria. It would be useful to conduct follow-up work to understand user perspective on what features to prioritize and how.

While we found fewer accessibility features provided within the tools themselves, we did notice a decent offering from third party plug-ins that either supported the creation or evaluation of accessible design. It is clear that Figma and Sketch have benefited from opening their platforms up to the design community to allow designers and developers to contribute new prototyping features. Figma had 17 accessibility plug-ins and Sketch had seven accessibility plug-ins both with a variety of features, whereas the other prototyping tools had only one to three plug-ins that provided basic color contrast checks and color blindness simulations.

However, if prototyping tools are relying on third party developers, rather than adding built-in accessibility features, then we do want to reflect on the potential limitations of this. If a specific accessibility feature is only available as a third party plug-in, then the user has to actively seek it out by searching through the plug-in store. A new user of the prototyping tool may not know about being able to install plug-ins. Finally, there is a question of validity. There is no clear vetting process and what determines a plug-ins success is likely community based (e.g., ratings, comments, feedback). The prototyping tool companies are potentially in a better position to develop accessibility features that accurately conform to current standards and best practice guidelines.

For future work, we will employ qualitative methods to interview different stakeholders (e.g., designers and prototyping tool companies) to understand their attitudes and concerns for built-in and plug-in accessibility features. We also plan to run additional evaluations of the accessibility features to understand how they are used within the design process and individual workflow styles to identify whether there are opportunities to improve usability.

4 CONCLUSION

Designers should include accessibility practices in the early phase of design such as while creating design prototypes. Prototyping tools can be used to provide good assistance to designers to verify design accessibility. We evaluated 10 design prototyping tools to research on the current accessibility assistance provided by these tools. There is support provided by prototyping tools largely in the form of third-party plug-ins, but minimal assistance in the form of built-in features. Also, the availability of these features varies from tool to tool. We argue that there is potential to improve accessible design by increasing the potential for accessibility to be consider earlier in the design process, but the current approach needs more refinement.

REFERENCES

[1] Shadi Abou-Zahra, Judy Brewer, and Shawn Lawton Henry. 2013. Essential Components of Mobile Web Accessibility. In *Proceedings of the 10th International*

Cross-Disciplinary Conference on Web Accessibility (Rio de Janeiro, Brazil) *(W4A '13)*. Association for Computing Machinery, New York, NY, USA, Article 5, 4 pages. https://doi.org/10.1145/2461121.2461138

[2] Apple. n.d.. Human Interface Guidelines: Visual Design. https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/. Accessed: 2020-07-18.

[3] Cynthia L. Bennett and Daniela K. Rosner. 2019. The Promise of Empathy: Design, Disability, and Knowing the "Other". In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300528

[4] Vicente Luque Centeno, Carlos Delgado Kloos, Martin Gaedke, and Martin Nussbaumer. 2005. Web Composition with WCAG in Mind. In *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A)* (Chiba, Japan) *(W4A '05)*. Association for Computing Machinery, New York, NY, USA, 38–45. https://doi.org/10.1145/1061811.1061819

[5] Lisa Dziuba. 2019. Accessibility tools for designers and developers. https://uxdesign.cc/accessibility-tools-for-designers-and-developers-ea400a415c0a. Last Accessed: 2022-6-11.

[6] Tânia Frazão and Carlos Duarte. 2020. Comparing Accessibility Evaluation Plug-Ins. In *Proceedings of the 17th International Web for All Conference* (Taipei, Taiwan) *(W4A '20)*. Association for Computing Machinery, New York, NY, USA, Article 20, 11 pages. https://doi.org/10.1145/3371300.3383346

[7] Google. n.d.. Material Design. https://material.io. Accessed: 2020-07-18.

[8] Samine Hadadi. 2021. Adee: Bringing Accessibility Right Inside Design Tools. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, USA) *(ASSETS '21)*. Association for Computing Machinery, New York, NY, USA, Article 101, 4 pages. https://doi.org/10.1145/3441852.3476478

[9] Shawn Lawton Henry and Andrew Arch. 2012. *Financial factors in developing a web accessibility business case for your organization.* W3C. http://www.w3.org/WAI/bcase/fin#decreasing. Accessed: 2014-08-16..

[10] Shayna Hodkin. 2019. 8 tools that make accessible design easier. https://www.invisionapp.com/inside-design/accessibility-tools/. Last Accessed: 2022-6-11.

[11] Gopinaath Kannabiran and Susanne Bødker. 2020. *Prototypes as Objects of Desire*. Association for Computing Machinery, New York, NY, USA, 1619–1631. https://doi.org/10.1145/3357236.3395487

[12] Andrew Kirkpatrick, Joshue O'Connor, Alastair Campbell, and Michael Cooper. 2018. Web Content Accessibility Guidelines (WCAG) 2.1.

[13] Chris Law, Julie Jacko, and Paula Edwards. 2005. Programmer-Focused Website Accessibility Evaluations. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility* (Baltimore, MD, USA) *(Assets '05)*. Association for Computing Machinery, New York, NY, USA, 20–27. https://doi.org/10.1145/1090785.1090792

[14] Jonathan Lazar, Patricia Beere, Kisha-Dawn Greenidge, and Yogesh Nagappa. 2003. Web accessibility in the Mid-Atlantic United States: a study of 50 homepages. *Universal Access in the Information Society* 2, 4 (2003), 331–341. https://doi.org/10.1007/s10209-003-0060-z

[15] Junchen Li, Garreth W. Tigwell, and Kristen Shinohara. 2021. Accessibility of High-Fidelity Prototyping Tools. 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 493, 17 pages. https://doi.org/10.1145/3411764.3445520

[16] Adriana Martín, Alejandra Cechich, and Gustavo Rossi. 2011. Accessibility at Early Stages: Insights from the Designer Perspective. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility* (Hyderabad, Andhra Pradesh, India) *(W4A '11)*. Association for Computing Machinery, New York, NY, USA, Article 9, 9 pages. https://doi.org/10.1145/1969289.1969302

[17] Microsoft. n.d.. Universal Windows Platform documentation. https://docs.microsoft.com/en-gb/windows/uwp. Accessed: 2020-07-18.

[18] Marian Padure and Costin Pribeanu. 2020. Comparing six free accessibility evaluation tools. *Informatica Economica* 24, 1 (2020), 15–25. https://doi.org/10.24818/issn14531305/24.1.2020.02

[19] Taylor Palmer and Jordan Bowman. 2020. 2020 Design Tools Survey. https://uxtools.co/survey-2020#conclusion. Last Accessed: 2022-6-11.

[20] K. Schneider. 1996. Prototypes as assets, not toys. Why and how to extract knowledge from prototypes. (Experience report). In *Proceedings of IEEE 18th International Conference on Software Engineering*. 522–531. https://doi.org/10.1109/ICSE.1996.493446

[21] Garreth W. Tigwell. 2021. Nuanced Perspectives Toward Disability Simulations from Digital Designers, Blind, Low Vision, and Color Blind People. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 378, 15 pages. https://doi.org/10.1145/3411764.3445620

[22] Garreth W. Tigwell, David R. Flatla, and Neil D. Archibald. 2017. ACE: A Colour Palette Design Tool for Balancing Aesthetics and Accessibility. *ACM Trans. Access. Comput.* 9, 2, Article 5 (Jan. 2017), 32 pages. https://doi.org/10.1145/3014588

[23] Shari Trewin, Brian Cragun, Cal Swart, Jonathan Brezin, and John Richards. 2010. Accessibility Challenges and Tool Features: An IBM Web Developer Perspective. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)* (Raleigh, North Carolina) *(W4A '10)*. Association for Computing Machinery, New York, NY, USA, Article 32, 10 pages. https://doi.org/10.1145/1805986.1806029

[24] Brian Wentz, Paul T Jaeger, and Jonathan Lazar. 2011. Retrofitting accessibility: The legal inequality of after-the-fact online access for persons with disabilities in the United States. *First Monday* 16, 11 (Nov. 2011). https://doi.org/10.5210/fm.v16i11.3666