Tangible Web Layout Design for Blind and Visually Impaired People: An Initial Investigation

Ashrith Shetty
University of Maryland
College Park, MD, USA
ashrith@terpmail.umd.edu

Ebrima Jarjue
University of Maryland
College Park, MD, USA
ebjarjue@terpmail.umd.edu

Huaishu Peng
University of Maryland
College Park, MD, USA
huaishu@cs.umd.edu

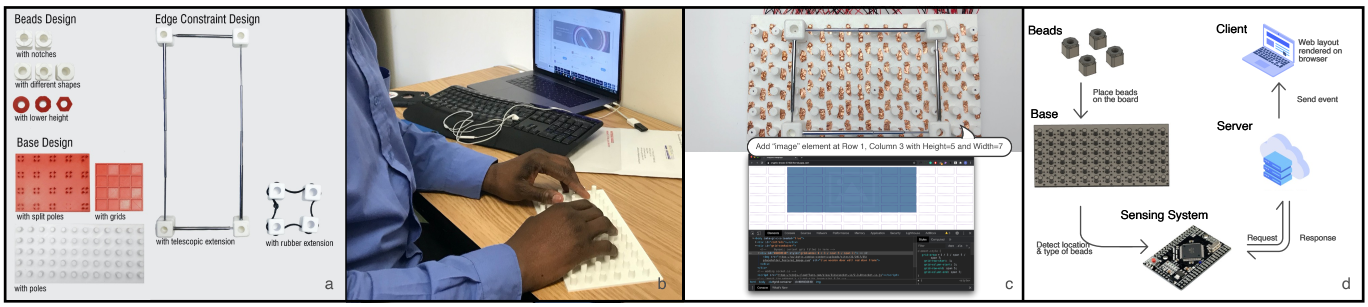

Figure 1. a) 3D printed artifacts, including different types of tangible beads, bases and constraints. b) Co-author Ebrima Jarjue interacting with one of the artifacts in the co-design session. c) First prototype of *Sparsha*, the web layout is rendered based on the placement of the beads d) System diagram.

ABSTRACT
In this poster, we explore the potential of using a tangible user interface (TUI) to enable blind and visually impaired (BVI) developers to design web layouts without assistance. We conducted a semi-structured interview and a co-design session with a blind participant to elicit insights that guided the design of a TUI prototype named *Sparsha*. Our initial prototype contains 3D printed tactile beads that represent HTML elements, and a 3D printed base that represents the web page layout. BVI users can add an HTML element to the layout by placing tactile beads on top of the base. The base senses the type and location of the beads and renders the HTML element in the corresponding location on the client browser. The poster concludes with a discussion and enumerates future work.

Author Keywords
Blind and visually impaired; tangible interface; accessible tools

CCS Concepts
•Human-centered computing → Human computer interaction (HCI); Accessibility systems and tools;

INTRODUCTION
The on-going democratization of assistive technologies is improving the quality of life for millions of people who identify as blind or visually impaired (BVI) [18]. Devices like screen readers, tactile printers, and braille displays have also led to new employment opportunities for BVI people [2, 4, 6]. Accessible programming tools such as *CodeTalk* [13] and *StructJumper* [3] make it possible for BVI developers to write code to express logic and algorithms. As a result, the number of blind programmers has been steadily rising in the last few years [14, 15]. Although BVI developers are now able to write code, they still lack accessible ways to define the visual layouts, for instance, the layout of an HTML webpage. Lacking direct feedback about the size and the position of elements, BVI developers can only write code to specify *what* content they create, but do not understand *where* to put them and *how* they look.

Recent works in this space [10, 12] address the *editing* of existing visual layout templates but do not support the creation of new layouts. While addressing similar problems, our approach is unique in that it enables BVI developers to create a visual layout from scratch while using their hands to physically manipulate the contents of the layout and without the assistance from sighted associates. We are inspired by ability-based design [19], an approach that advocates shifting the focus of accessible design from disability to ability. Designers should strive to leverage all that their target audience *can* do [19]. Previous studies have revealed that BVI people have higher tactile acuity than sighted people [5] and use their sense of touch to "navigate and negotiate the world" [19]. Hence, we see value in a tool that can support this tactile perceptual ability by using a Tangible User Interface (TUI) [7].

Employing a user-centered research method, we started with a semi-structured interview to understand the challenges BVI people face as well as potential solutions that they have explored (if any). Based on the interview, we prepared a list of

UIST '20 Adjunct, October 20–23, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7515-3/20/10.
http://dx.doi.org/10.1145/3379350.3416178

low-fidelity physical probes and artifacts that can potentially help BVI developers in layout creation. These physical probes were then used in a co-design session with the BVI participant. The behavioral evidence collected from the workshop, as well as the interview with the user guided the design of our initial prototype *Sparsha*. Below we will introduce each of the research activities, the main features of *Sparsha*, and conclude with a list of future research activities.

FORMATIVE STUDY

Co-author Ebrima Jarjue, a blind graduate student who is also a researcher in the field of Accessibility, was the participant for the interview as well as the co-design session [17]. We conducted an hour-long, semi-structured interview [1] with him to find out how he interacts with websites and about his prior experience with designing or creating any visual layouts like websites or slide decks. We used an interview guide with open-ended questions [1] to get more descriptive answers.

DESIGN CRITERIA

Based on the insights gained from the interview, we came up with the following design criteria for the tangible layout design tool.

1. The tool should allow users to specify a location on the screen where they want to add the required HTML element.
2. BVI users have no way of knowing how their actions translate visually on the screen. Hence tactile representations of the web page layout and HTML elements will allow BVI users to use their fingers to add HTML elements as well as to locate and modify elements that have already been added.
3. Users need to be able to add different kinds of HTML elements on the TUI. The type, location, and size of the elements placed on the tactile surface need to be translated onto the screen accurately.
4. Users need to be given feedback that the action was completed successfully.

CO-DESIGN

Based on the interview and the distilled design criteria, we generated three types of haptic beads, two types of constraints for the beads, and three types of bases as design artifacts (Figure1a). These artifacts vary in size, shape, and fidelity in terms of tangible feedback. We presented all artifacts at the co-design [17] session with Ebrima to understand how they may help BVI developers to create a visual layout. We let him get a feel for them one by one, followed by a side by side comparison. He was encouraged to think aloud [8, 11] throughout the session and to discuss the pros and cons of each artifact design and also suggest design improvements that would make the solution more effective. He was also asked to perform tasks such as finding a specified location on the base by feeling the tactile surfaces, and placing four beads in a rectangular shape on it at a specified location (Figure1c). The behavioral evidence was collected and observed [9].

Based on the co-design session, we decide that our prototype of *Sparsha* will consists of a pole base that represents the web page. Sets of four cuboid-shaped haptic beads represent rectangular HTML elements. The beads would be constrained by a telescopic constraint mechanism to allow for resizing. The beads would also have haptic shapes on the top in order to differentiate the different types of beads to represent different HTML elements.

SPARSHA: A TANGIBLE LAYOUT DESIGN TOOL

We now introduce *Sparsha*, our initial prototype to support tangible layout design. The major components of *Sparsha* includes a tactile base that represents the screen real estate, a set of tactile beads acting as HTML elements of different types, and a web application gets inputs from the hardware and renders the HTML layout accordingly (Figure1c). We chose to go with a 12 column CSS Grid implementation for the layout design since that is one of the most commonly used and is the most flexible for accommodating different kinds of layouts.

The system diagram is shown in Figure 1d. When a BVI user places four beads of a particular content type on the board, say of image type, the location and type of the beads are sensed by its Arduino-based sensing circuit. The beads would be in the shape of a rectangle to represent the rectangular shape of HTML elements. The location of the beads provides the size of the required image element. Resistors of different values are placed inside the beads to differentiate between beads representing different HTML elements. The Arduino then calls an API using the attached Wifi module and passes the type of the beads (image), the coordinates of the top-left corner of the image, as well as the size of that image as the parameters. The Web server receives this information and sends an event to the client browser to render an image of the given size and at the given location to the client browser. The image element is then rendered on the screen, with a stock image pulled using the Unsplash API [16], along with audio feedback stating that the image element was added at the specified location.

One of the limitations of the current prototype tool is the size of the board and the screen area that can be represented at a time. Currently we are limited by the size of the 3D printers. The current version also does not support overlapping HTML elements because of the way the telescopic-extension constraints on the beads are built. Another limitation of this prototype is that it is focused on making the creation of new layouts easy but does not have a way to edit existing layouts. The prototype also does not support editing of content at the moment.

CONCLUSION

In this research, we present an exploratory attempt at designing and developing a novel, accessible tool *Sparsha* that supports BVI to design visual layout using tangible user interface. We interviewed a blind graduate student and conducted a co-design session with him to gain useful insights which formed the basis of the prototype. In the next steps, we plan to conduct a formal evaluation with BVI participants to understand its effectiveness, and then work on incorporating content addition and management as the next feature.

REFERENCES

[1] William Adams. 2015. Conducting Semi-Structured Interviews. DOI: http://dx.doi.org/10.1002/9781119171386.ch19

[2] Julia Anderson. 1989. How Technology Brings Blind People into the Workplace. (1989). https://bit.ly/2OI6jYZ (Last accessed on 03/09/2020).

[3] Catherine M Baker, Lauren R Milne, and Richard E Ladner. 2015. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3043–3052. DOI: http://dx.doi.org/10.1145/2702123.2702589

[4] Edward Bell and Natalia Mino. 2015. Employment Outcomes for Blind and Visually Impaired Adults. *Journal of Blindness Innovation and Research* 5 (1 2015). DOI: http://dx.doi.org/10.5241/5-85

[5] Zaira Cattaneo and Tomaso Vecchi. 2011. *Blind Vision: The Neuroscience of Visual Impairment*.

[6] Parham Doustdar. 2016. The Tools of a Blind Programmer–Parham Doustdar's Blog. (2016). https://www.parhamdoustdar.com/2016/04/03/tools-of-blind-programmer/ (Last accessed on 03/09/2020).

[7] Hiroshi Ishii. 2008. *Tangible bits: Beyond pixels*. DOI: http://dx.doi.org/10.1145/1347390.1347392

[8] Anker Helms Jørgensen. 1990. Thinking-aloud in user interface design: A method promoting cognitive ergonomics. *Ergonomics* 33 (4 1990). DOI: http://dx.doi.org/10.1080/00140139008927157

[9] Clayton Lewis and John Rieman. 1993. Task Centered User Interface Design: A Practical Introduction. (4 1993).

[10] Jingyi Li, Son Kim, Joshua Miele, Maneesh Agrawala, and Sean Follmer. 2019. *Editing Spatial Layouts through Tactile Templates for People with Visual Impairments*. 1–11 pages. DOI: http://dx.doi.org/10.1145/3290605.3300436

[11] Erica Olmsted-Hawala, Elizabeth Murphy, Sam Hawala, and Kathleen Ashenfelter. 2010. Think-aloud protocols: A comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability. In *Conference on Human Factors in Computing Systems - Proceedings*, Vol. 4. 2381–2390. DOI: http://dx.doi.org/10.1145/1753326.1753685

[12] Venkatesh Potluri, Liang He, Christine Chen, Jon Froehlich, and Jennifer Mankoff. 2019. *A Multi-Modal Approach for Blind and Visually Impaired Developers to Edit Webpage Designs*. 612–614 pages. DOI: http://dx.doi.org/10.1145/3308561.3354626

[13] Venkatesh Potluri, Priyan Vaithilingam, Suresh Iyengar, Y Vidya, Manohar Swaminathan, and Gopal Srinivasa. 2018. CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA. DOI: http://dx.doi.org/10.1145/3173574.3174192

[14] Stack Overflow 2018. Stack Overflow Developer Survey 2017. (2018). https://insights.stackoverflow.com/survey/2017 (Last accessed on 03/09/2020).

[15] Stack Overflow 2020. Stack Overflow Developer Survey 2019. (2020). https://insights.stackoverflow.com/survey/2019 (Last accessed on 03/30/2020).

[16] Unsplash Image API 2020. Free HD Photo API. (2020). https://unsplash.com/developers (Last accessed on 04/01/2020).

[17] Maja Velden and Christina Mörtberg. 2014. Participatory Design and Design for Values. In *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains*. 1–22. DOI: http://dx.doi.org/10.1007/978-94-007-6994-6_33-1

[18] WHO 2019. Blindness and Vision Impairment. (2019). www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (Last accessed on 07/10/2020).

[19] Jacob O Wobbrock, Shaun K Kane, Krzysztof Z Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-Based Design: Concept, Principles and Examples. *ACM Trans. Access. Comput.* 3, 3 (4 2011). DOI: http://dx.doi.org/10.1145/1952383.1952384