UML Modeling for Visually-Impaired Persons

Brad Doherty and Betty HC Cheng
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824, USA
{dohert72, chengb} at cse.msu.edu

***Abstract*—Software modeling is generally a collaborative activity and typically involves graphical diagrams. The Unified Modeling Language (UML) is the *de facto* standard for modeling object-oriented software. It provides notations for modeling a system's structural information (e.g. databases, sensors, controllers, etc.), and behavior, depicting the functionality of the software. Because UML relies heavily on graphical information, visually impaired persons (VIPs) frequently face challenges conceptualizing the often complex graphical layouts, involving numerous graphical objects. The overall objective of the PRISCA project is to facilitate collaborative modeling between VIPs and other project team members. Towards this end, this paper describes preliminary PRISCA work into developing software that automatically generates a haptic 3D representation of the UML diagrams from the output of an existing UML diagram editor. In addition, textual annotations for the models are converted to Braille and printed in 3D atop the respective graphical objects. Research and human factor challenges are reviewed in the paper in an effort to raise the level of awareness to the MDE community of this important area of work.**

I. Introduction

While UML and other graphical modeling approaches have become key components of the modern software design process, the heavy use of visual features as descriptions has created unintentional obstacles for software developers with visual impairments. Recreating the diagrams as haptic models has demonstrated a viable solution for addressing this problem [1], [2] (e.g., using pushpins and string on a sheet of cardboard to represent graphical objects), however, an automatic method for producing such models directly from the editing source used to create the models for sighted developers does not exist. Moreover, requiring software developers to recreate the models in a 3D modeling environment is not feasible given time and effort constraints imposed by project deadlines. This paper describes PRISCA, a tool-chain that automatically renders a 3D-printable haptic UML model from an existing UML modeling tool. [1]

[1]This project is inspired in part by Priscilla McKinley, a VIP who was an advocate for making computing and technology accessible to VIPs [3].

Presenting a 2D visual object to an individual with a visual impairment can be difficult as verbal-based descriptions become less effective as the complexity of an image increases. The TeDUB software [4], [5], [6] automates the description process by translating a UML diagram into a navigable XML type format that can be processed by a standard text reader, thus providing a more efficient process for describing models to VIPs than verbal-based descriptions [7], [8]. A drawback of such a tool, however, is its inability to fully capture visually-dependent descriptions, such as relative spacing of diagram components, relationships depicted by differing connector types, etc., all of which are features commonly used in UML and other software modeling. The use of haptic translations of such models is one approach to overcoming such challenges. A system of cardstock, plastic strips, and pins developed by Brookshire [2] demonstrates the usefulness of haptic models. Brookshire's method however, requires significant effort from a sighted person to manually assemble the models, which is usually not feasible under time and effort constraints. Automatically creating haptic displays has seen some success with a refreshable tactile pin display developed by Roberts *et al.* [9]. This system has the added benefit of being able to easily produce Braille descriptions along with the tactile graphic, but the low resolution of the pins is a limiting factor.

This paper presents preliminary results for PRISCA, a project whose overall objective is to facilitate software modeling collaboration between VIP developers and other developers by using haptic representations of software models. Specifically, PRISCA currently supports a proof of concept tool-chain that takes output from an existing UML modeling tool and transforms the models into corresponding 3D representations defined in terms of the input language for an existing 3D printer. PRISCA also generates a 3D Braille representation of the textual annotations included in the graphical models. In addition, we have reusability and extensibility as guiding principles when developing the PRISCA translation and rendering components in order to handle multiple types of diagrams.

An overarching goal is to make the technology accessible, without requiring developers to purchase expensive software or hardware, or incur development time to recreate software models in a 3D format. As such, PRISCA automatically processes the XML output from *Visual Paradigm* [10], a commonly-used commercial modeling tool, to produce an STL (STereoLithography) file, the input language for the *MakerBot*® 3D printer ("affordable and powerful consumer 3D printer" [11]). The tool-chain incorporates techniques to develop 3D representations of the diagram element using 3D modeling libraries. PRISCA also includes a utility for translating the textual annotations (e.g. class name and attributes) into a 3D representation of the corresponding Braille text. Thus far, as proof of concept of the approach and given their frequent use in practice, we have focused on producing UML class and sequence diagrams (including example Braille text annotations) to capture structural and behavioral information, respectively.

Our tool chain has been validated on sample models created with *Visual Paradigm UML* [10], including models created for projects developed in collaboration with industrial partners. The remainder of this paper is organized as follows. Based on the literature and our experiences with VIP students, Section II highlights three key obstacles to making graphical artifacts accessible to VIPs. In Section III, we overview the approach we used to develop PRISCA, including the key elements of PRISCA and examples of PRISCA renderings. Next, in Section IV, we discuss open issues and research challenges with developing modeling support for VIPs through our work with PRISCA. Section V summarizes the work and discusses future investigations.

II. Position Statement

Based on review of the literature and feedback from VIP students studying computer science and software engineering, three key obstacles make graphical artifacts difficult to access by visually-impaired persons (VIPs).

A. Technology Limitations

First, there exists limited technology for translating graphical artifacts into an accessible form for VIPs. Three complementary approaches have been developed to provide modeling support for VIPs: text-based descriptions, smart interfaces with voice-over textual descriptions, and haptic recreations of models. Each of these are briefly reviewed, including their limitations.

The TeDUB diagram interface [7] relies on verbal descriptions of diagrams to convey the contents to visually impaired persons. The process begins with the input of a bitmap image of a diagram [5]. TeDUB then translates the bit map image into a semantically enriched format and uses this XML-like structure to produce a navigable hierarchy that is accessible through an interface [5]. This tool relies on automatic recognition of nodes and other hierarchical structures found on the diagram [7]. Upon analysis of the system, all the users in a study found the software provided a sufficient way to learn how to read the UML diagrams, where a few criticisms were noted regarding the representation of hierarchical structured diagrams [5].

Another approach is to provide audio translations of the diagrams directly. Research and pedagogical activities (e.g., PLUMB [12], [13]), along with companies like ViewPlus [14] and TouchGraphics [15] have made progress in creating audio translations of graphical artifacts. Smart devices, such as the Apple® iPAD® [16] have voiceover features that can also provide audio descriptions of graphical artifacts. As with the text-based presentation of the diagrams, it can be challenging to establish spatial and hierarchical relationships between graphical icons and maintain a conceptual model as it evolves. Unfortunately, without a mental model of a given graphical artifact, it can be extremely challenging to fully comprehend the syntax and the intended semantics of the models, despite detailed text-based descriptions [17]. To this point, the National Federation of the Blind (NFB) has emphasized that while text-based approaches to modeling are helpful in making graphics more accessible to VIPs, the lack of tactile interfaces is still a significant limiting factor [18].

Limited work has been explored to develop tactile representations, but they have been shown to be effective in describing graphical relationships, including hierarchical structures. One such example was used to teach database diagrams to visually impaired students [2]. This approach used cardstock cards to represent classes, plastic strips for connectors, and push pins to describe cardinality. The objects were fastened to a cardboard mounting surface, thus providing a haptic diagram representation. Upon evaluation, students admittedly appreciated the tactile design over an auditory graphics system, and collaboration with sighted students was seen as an advantage of such a design [2]. This system is limited by the amount of manual assembly required, thus making it less feasible in an environment constrained by time and effort. A similar approach was used to teach UML class diagrams at Michigan State University. Specifically, a sheet of cardboard with push pins and string were used as a means to "display" a simple model to a VIP in the Computer Science and Engineering (CSE)

Department [1]. While these approaches might be useful for illustrating a specific model, it is clearly not scalable nor practical even for a collaborative semester project, involving sighted developers.

Automatic tactile displays can be used to decrease the amount of effort required to perform manual assembly of diagrams. The use of a refreshable tactile graphic display described by Roberts *et al.* [19], provides a reusable surface for rendering haptic displays. A graphical image of the model was scanned to produce a sequence of actuation signals for a "bed of nails" to represent the graphical artifacts images [20]. Specifically, the system used an array of pins that could be raised or lowered individually to create an image in a similar mode to pixels on a screen. The resolution was limited, where only 3600 nails were used as actuators. These displays can be navigated by sense of touch, much like the card and plastic strip system described previously, providing a similar tactile medium for diagram descriptions. Refreshable tactile displays are limited by the detail that they can provide however, as the pin density is ten pins per inch [19], which may cause difficulties in providing optimal resolution for some diagrams.

B. Project Resource Constraints

A second obstacle is the limited resources in terms of time and money, which prohibit collaborators from redrawing models produced in a collaborative modeling environment using a 3D modeling tool as a means to produce tactile output representations of the models. In order for the VIP to contribute to modeling activities, they must be working with the same version of the model as the sighted team members. If major changes are made to the models, then those changes have to be recreated and updated in the 3D modeling tool. This additional burden will make it unattractive to participate in such a team for both the sighted and the VIP team members. Eventually, the VIPs role in the modeling efforts will decrease as the project progresses.

C. Misperceptions

The third obstacle is the lack of education and misperception of the capabilities of VIPs to contribute to modeling [21]. Studies have shown that when one of our main senses is lost, the remaining senses are heightened and even more developed than those without the disability [22]. For people with visual disabilities, their auditory, tactile, and language processing skills are particularly advanced [21]. As such, by making existing modeling platforms accessible to VIPs, we gain access to these people and their skills that would otherwise not be available to a team. Despite tenacity and their pioneering spirit, many VIPs have encountered the above-mentioned obstacles [23], [24], [25].

III. APPROACH

This section describes the PRISCA project. We start by overviewing our objectives in developing PRISCA. Then we describe the enabling technologies and process that we used to support the PRISCA tool-chain. Then we demonstrate the use of PRISCA on examples and highlight example human factor issues that motivated our specific design decisions.

A. Objectives

The main objective of PRISCA is to facilitate VIP collaboration in software modeling. PRISCA does this by automatically translating the output of a UML diagramming tool into a 3D printer format by leveraging and extending existing parsing and 3D rendering libraries. PRISCA also handles the translation of textual annotations within a UML diagram by translation into a 3D Braille format atop the graphical elements. Priorities in the design of PRISCA include a focus on reusability of diagram features (e.g. similar shapes and line types), extensibility to other diagram types, and finally extensibility to a range of modeling tools and 3D printers.

B. Modeling and Printing Facilities.

As mentioned before, an overarching goal of this project is accessibility of the technology to VIPs. To this point, we selected a relatively easily accessible modeling tool and affordable 3D printer. Specifically, we selected *Visual Paradigm* [10] given its differential licensing program, with special consideration for academic institutions, its XML output format, its availability for use on multiple platforms, and its support for multiple diagramming notations in addition to UML.

For producing the 3D representations of the 2D UML diagrams, we make use of a graphics library from OpenSCAD, an open source computer aided design software commonly used to create 3D CAD models [26]. OpenSCAD is built atop multiple software libraries and is available for use on multiple platforms. As such, it provides a rich set of primitives that we can use, including a scripting language to create 3D representations of the 2D UML diagram elements.

In order to produce the physical 3D, haptic representation of the model, we use the MakerBot® Replicator 2 3D printer. This printer is affordable by individual users (i.e., less than $3000 USD), and it is becoming increasingly popular in academic settings for students and faculty to use. Of particular interest for our project, it has an option to accept input in an ASCII representation of the STL (STereo Lithography), a commonly used language in many software packages for 3D printing, CAD modeling, and rapid prototyping. A key feature for this project is its simplicity in that it only represents surface geometry of a 3D object, without support for color or texture, neither of which is needed for our current purposes.

C. Overview

This section overviews the PRISCA approach. Based on classroom and industrial experiences, our intent is for a development team to use *Visual Paradigm* [10] to create a UML diagram and then export it as a standard XML file. PRISCA processes the XML file to produce an STL file that can be sent to a *MakerBot*® 3D printer to produce a 3D, haptic representation of the UML diagram.

Figure 1 gives a data flow diagram (DFD) of the PRISCA tool chain. Here, rectangles denote external entities, circles describe processes, two parallel lines delimit a grammar file (i.e. persistent data), and arrows represent data flows. Starting with the modeling tool Visual Paradigm, a UML diagram can be exported as an XML document (1). PRISCA parses the XML file to produce the corresponding 3D representations of the graphical UML objects comprising rendering instructions defined in terms of the OpenSCAD library utilities. Then PRISCA uses the OpenSCAD utility to generate the ASCII STL format (2) from the OpenSCAD instructions. The STL file generated by PRISCA can then be sent to a 3D printer to render the final diagram (3).

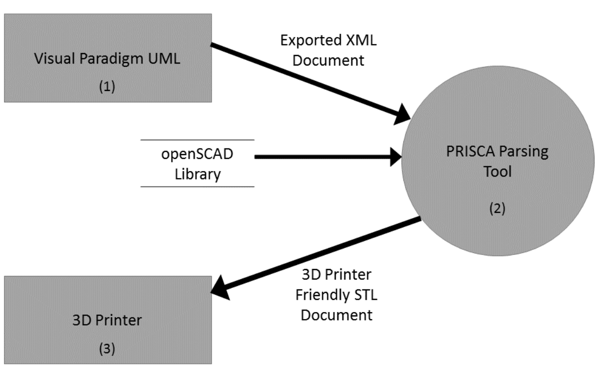

Fig. 1. DFD for PRISCA tool-chain.

D. Example Diagrams

The following section provides examples of UML diagrams processed by PRISCA. In order to produce the optimal contrast in the image, the STL as viewed in OpenSCAD's viewer has been used in lieu of a photographic image of the physical 3D "print-out". Figure 2 shows a UML class diagram created in *Visual Paradigm* (in blue) and the PRISCA rendering below (in yellow). Figure 3 shows a *Visual Paradigm* sequence diagram (in blue) and its PRISCA rendering (in yellow).

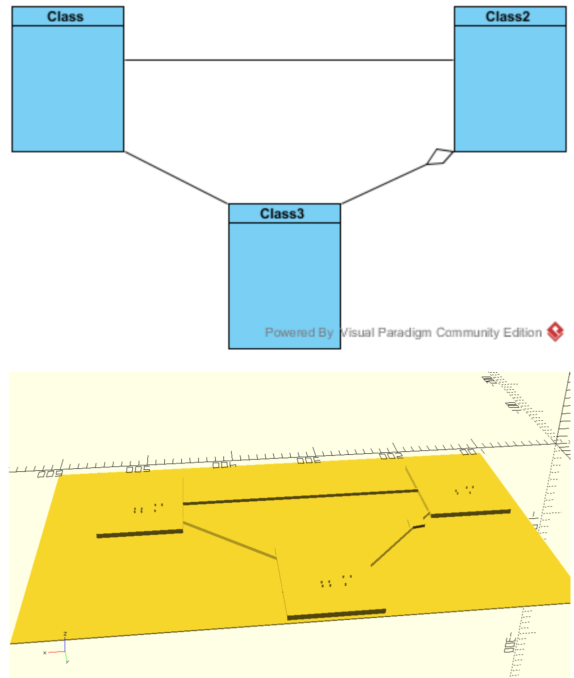

Fig. 2. UML class diagram (top) and PRISCA rendering (below).

E. Human Factor Issues

These examples help to illustrate the challenges we faced when trying to teach UML diagramming to a VIP student in a project course involving industrial collaborators. In an earlier course, the student had previously used the cardboard with push pins and string to learn the syntax of a class diagram [1]. With that haptic-based information, she was able to actively contribute to the discussions associated with the class diagram. But the sequence diagram was taught only through oral instruction (using descriptive explanations of shapes, connectors, layouts, etc.) and from textbook descriptions. Based on a debriefing session with the student after the course completed, despite the simplicity of the sequence diagram notation, the student never gained a proficient understanding of the object lifelines and the messages between them. As a result, the student was only able to contribute at a conceptual level to discussions related to

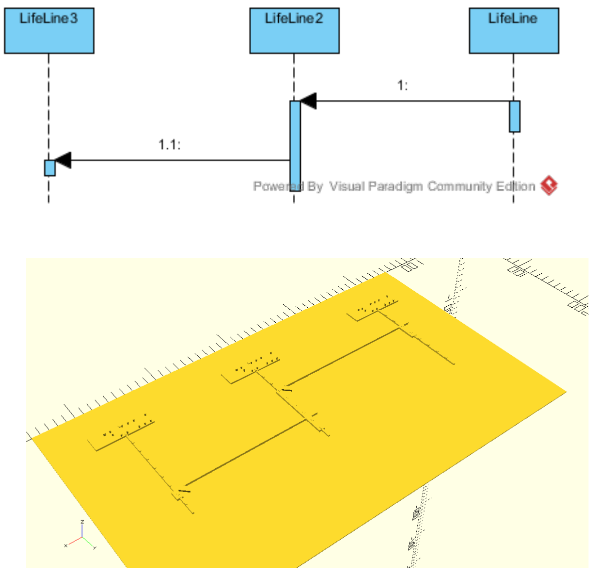

Fig. 3. UML sequence diagram (top) and PRISCA rendering (below).

sequence diagram modeling. The sequence diagram information seemed abstract, lacking concrete connection to the project (including the class diagram elements). Now with PRISCA, the student has access to a haptic representation of the sequence diagram and can create a mental model of the spatial relationships between the graphical elements. Furthermore, PRISCA enables VIPs to obtain access to models (and their revisions) without incurring any additional modeling efforts from the rest of the development team.

A key human factor issue that was raised with this experience is how to effectively communicate with VIPs (or others with disabilities). In part due to lack of lead time in realizing that a VIP was taking the course (i.e., the VIP student approached the instructor at the end of the first day of class to ask for an electronic copy of the syllabus and the other materials that were distributed to the class), it was not possible to rewrite the course materials to be more descriptive for a VIP. As such, , an attempt was made to add impromptu descriptive explanations during the live delivery of lectures. The VIP student was asked at the end of each lecture if there were any questions regarding the lecture material, or if additional examples could be reviewed with the VIP to illustrate the modeling syntax and/or semantics. In each case, the VIP declined any additional assistance. During the debriefing after the conclusion of the course, the VIP student informed the instructor that she thought that she understood the modeling materials sufficiently, but only when taking an exam containing questions about the models did she realize that she really did not understand the materials. She had not gained an understanding of the spatial relationships between the graphical elements of the new modeling notations presented in the class. When asked why she did not ask for help, she responded that she did not want to be a burden on the team or the instructor and felt that she could "figure it out". (She also conveyed a few negative experiences in previous classes, where instructors refused to help her with modeling-related topics since they did not feel it was worth their time given her visual impairment.)

With the team project deliverables, she was assigned tasks that did not deal directly with the models (e.g., she was the webmaster for the project team, where their team was the only one who created a webpage that satisfied all the content criteria; she wrote several sections of the requirements specification document; and she contributed to the coding of the prototype). Through peer reviews, it was clear that she was a highly valuable team member, performing all assigned tasks with the utmost quality in a timely fashion.

It is for people like this VIP student and other highly talented and motivated VIPs who have much to contribute to research and teaching activities involved with modeling that the PRISCA project draws its inspiration.

IV. Open Issues and Research Challenges

This section discusses several challenges uncovered during the course of the development of PRISCA.

Handling Textual Annotations: Handling the textual annotations found on UML diagrams has been a challenge in the development of PRISCA (e.g., association labels, class attributes and operations), where we are continuing to investigate better ways to provide the textual descriptions in a way that both assists VIPs and supports collaboration. The original plan for the textual annotations was to convert the text to Braille and print the 3D model with Braille descriptions presented in the same manner as the original diagram's textual annotations. The main challenge with translating the text directly to Braille is that the minimum size of Braille text is too large relative to the 3D print area, thus not allowing the same amount of textual description found in a standard UML diagram (see Figure 4). While this limitation might ultimately rule out translation to and production of all the diagram annotations in Braille, the most effective medium for relaying textual annotations will be determined through interactive feedback with VIPs and further investigation of emerging technologies.

Representing Connector Endpoint Information: A key advantage to diagramming, particularly well-illustrated with the UML class diagram, is the ability

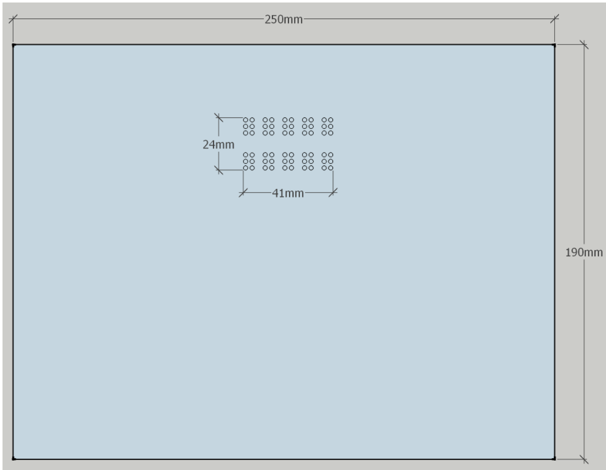

Fig. 4. To-scale print area for a typical 3D printer overlaid with ten Braille characters of minimum size and spacing. Notice the large amount of space that each character requires.

to represent a rich amount of information in a concise fashion. For example, consider the potential pieces of information that can be represented with a binary association between two classes (e.g., association label, multiplicities, link attributes, role names, directed association, aggregation, etc.) Positioning connector endpoints more effectively, optimizing connector thickness, and determining the best method for scaling the diagram also present challenges for PRISCA development. These challenges are being addressed through trial and error as we continue to apply PRISCA to a variety of diagram layouts and through extending the PRISCA translation procedures.

Applicability and extensibility of PRISCA: As we continue to explore more diagramming notations, we may discover that the current graphics library/utilities are insufficient to capture the level of details needed. Or the STL language may be insufficient to capture the level of sophistication needed. For example, different textures and/or thickness variations may be one way to capture coloring schemes commonly used in diagramming tools.

Another challenge is how to make our techniques portable to other modeling tools. While XML is intended to be a standard output format for UML modeling languages, we have found that the XML output from one UML modeling tool is not directly usable by another UML modeling tool. Previously, we explored the development of an “XML-interchange” language that could be a common format to which various vendor-specific or tool-specific XML languages could be mapped.

Finally, while the current work applies to UML modeling, our longer term goal is to extend the capabilities to other commonly-used diagramming notations, such as those used for mathematical graphing, MATLAB, etc.

Collaborative Modeling: As mentioned earlier, VIPs have heightened senses to compensate for the visual impairment, such as memory and touch. We are continuing to explore the most effective process for using PRISCA to produce 3D representations of the UML diagrams. In particular, what degree and types of model changes warrant generating an updated haptic representation. During the course of creating a model for a software project, the beginning stages might involve dramatic changes from one iteration to the next as the development team gains a better understanding of the requirements and design constraints from the customer. As the project progresses, the changes become more fine-grained and occur less frequently. We will work with VIPs to determine what is the most effective means to convey the changes in order to enable them to effectively provide feedback on the models.

V. Conclusion

This paper describes PRISCA, a proof of concept project to make progress in enabling VIPs to collaborate with other (sighted) developers by working with 3D “print-outs” of software models. Automatically creating haptic 3D diagrams from the tools used by other developers enables PRISCA to be practically implemented in a setting constrained by time and effort. Currently, it takes about 1.5-2 hours to produce a 3D representation of a UML diagram with 10-15 elements. (The upper time-frame is needed when producing the sequence diagrams since the object lifelines and their labels are printed atop a supporting sheet of plastic; the 3D “print-out” of the diagram is too fragile to handle otherwise.) The conversion of text to Braille further facilitates comprehension of a model.

The original motivation for the objective of PRISCA was to teach UML modeling to visually-impaired computer science students to enable their collaboration on industry-sponsored team projects. As such, introduction to the syntactic elements and their (spatial) relationships was an important capability. As with sighted developers, proficiency with modeling for students and industry staff will come with practice and exposure to a variety of examples. PRISCA enables VIPs to gain at least a cursory understanding of UML modeling notations beyond what might be gained from auditory and/or textual description-based instruction.

Our future work will pursue several dimensions of the PRISCA project, including the aforementioned research challenges. The focus on reusability of diagram elements facilitates our ability to extend PRISCA to handle other diagram types, beyond the sequence and class UML diagrams presented in the paper. We will continue to work with VIP developers to expand PRISCA to address the mentioned research challenges and to facilitate modeling collaboration between VIPs and other development team members. We will also explore and leverage the emerging technology associated with the use of haptic representations for images on websites [27], innovative user interface designs specifically for VIPs [28], and 3D art with touch sensors that trigger auditory descriptions and explanations [29].

ACKNOWLEDGMENTS

This work has been supported in part by NSF grants CNS-1305358 and DBI-0939454, Michigan State University EnSURE Program, General Motors Research, and Ford Motor Company. The authors gratefully acknowledge the early work with the OpenSCAD libraries for the PRISCA project done by Marcos Botros. Jordyn Castor has been an inspiration for the project, answering questions and providing feedback for the PRISCA project. Finally, we greatly appreciate the detailed comments provided by the reviewers.

REFERENCES

[1] C. B. Owen, S. Coburn, and J. Castor, "Teaching modern object-oriented programming to the blind: An instructor and student experience," in *Proceedings of 2014 ASEE Annual Conference*, (Indianapolis, Indiana), ASEE Conferences, June 2014. Available from website https://peer.asee.org/23100.

[2] R. G. Brookshire, "Teaching UML database modeling to visually impaired students," *Issues in Information Systems*, vol. 7, no. 1, pp. 98–101, 2006.

[3] P. L. McKinley, "Literacy in the Lives of the Blind: An ethnographic study in the San Francisco Bay area." PhD Dissertation, 2006.

[4] H. Petrie, C. Schlieder, P. Blenkhorn, D. G. Evans, A. King, A.-M. O'Neill, G. T. Ioannidis, B. Gallagher, D. Crombie, R. Mager, and M. Alafaci, "Tedub: A system for presenting and exploring technical drawings for blind people," in *Proceedings of the 8th International Conference on Computers Helping People with Special Needs*, ICCHP '02, (London, UK, UK), pp. 537–539, Springer-Verlag, 2002.

[5] M. Horstmann, M. Lorenz, A. Watkowski, G. Ioannidis, O. Herzog, A. King, D. G. Evans, C. Hagen, C. Schlieder, A.-M. Burn, N. King, H. Petrie, S. Dijkstra, and D. Crombie, "Automated interpretation and accessible presentation of technical diagrams for blind people," *New Review of Hypermedia and Multimedia*, vol. 10, no. 2, pp. 141–163, 2004.

[6] A. King, P. Blenkhorn, D. Crombie, S. Dijkstra, G. Evans, and J. Wood, *Presenting UML software engineering diagrams to blind people*. Springer, 2004.

[7] D. Miller, "Accessible diagram interfaces," 2006. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.6892&rep=rep1&type=pdf.

[8] D. Miller, *Can we work together?* PhD thesis, University of North Carolina, 2009.

[9] W. M. H. Oosting, "Giving Visual Impaired People an Extra Dimension: Designing a Tactile Graphics Handling Braille Display," in *1st Twente Student Conference on IT*, 2004.

[10] Visual Paradigm, http://www.visual-paradigm.com/features, *Visual Paradigm Quick Start*, 12.1 ed., April 2015.

[11] CNET Staff, "CNET MakerBot Replicator Review," 2012.

[12] R. F. Cohen, R. Yu, A. Meacham, and J. Skaff, "PLUMB: Displaying Graphs to the Blind Using an Active Auditory Interface," in *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 182–183, ACM, 2005.

[13] R. F. Cohen, A. Meacham, and J. Skaff, "Teaching graphs to visually impaired students using an active auditory interface," in *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, pp. 279–282, ACM, 2006.

[14] ViewPlus, "Delivering Sense Ability." Tiger Software Suite for translating text to Braille. Website: https://viewplus.com/software/.

[15] TouchGraphics, "Tactile Design for Universal Access."

[16] Apple Inc., "Chapter 11. Using VoiceOver Gestures." Description of features on website: https://www.apple.com/voiceover/info/guide/_1137.html.

[17] J. Castor, "Experiences with modeling, from high school to college." Personal Communication, Fall 2014.

[18] D. Brauner and E. Summers, "iOS Accessibility: Teaching the Teachers," 2013. Retrieved from website: https://nfb.org/images/nfb/publications/fr/fr32/1/fr320112.htm.

[19] J. Roberts, O. Slattery, J. O'Doherty, and T. Comstock, "37.2: A New Refreshable Tactile Graphic Display Technology for the Blind and Visually Impaired," *SID Symposium Digest of Technical Papers*, vol. 34, no. 1, pp. 1148–1151, 2003.

[20] National Institute Of Standards And Technology, "Nist "Pins" down imaging system for the blind," *ScienceDaily*, vol. 13, November 2002.

[21] National Federation of the Blind of Washington, "About blindness." Retrieved from website. http://www.nfbw.org/blindness.html.

[22] M. Bates, "Super powers for the blind and deaf: The brain rewires itself to boost the remaining senses," *Scientific American*, September 2012.

[23] "Priscilla McKinley Obituary." Mitchell County Globe Gazette, December 2010. http://globegazette.com/mcpress/obituaries/priscilla-mckinley/article_d60b3c1e-0c8a-11e0-9c41-001cc4c002e0.html.

[24] P. L. McKinley, "Baby Steps, Long Strides, and Elephant Seal Humps." Braille Monitor, 2004. Available from website: https://nfb.org/images/nfb/publications/bm/bm04/bm0410/bm041003.htm|.

[25] J. Castor, "Pursuing a Dream and Beating the Odd." Michigan State University Newsletter. Retrieved from website: http://givingto.msu.edu/stories/story.cfm?id=33.

[26] M. Kintel, *About OpenSCAD*. OpenSCAD, Retrieved from http://www.openscad.org/about, 2015.03 ed., March 2015.

[27] A. Bajcsy, Y.-S. Li-Baboud, and M. Brady, "Depicting web images for the blind and visually impaired." SPIE website, 2013. Retrieved from http://spie.org/x104896.xml.

[28] D. Schnelle-Walka and M. Mhlhäuser, "User interfaces for brainstorming meetings with blind and sighted persons." Darmstadt University website. Available at https://www.tk.informatik.tu-darmstadt.de/de/research/talk-touch-interaction/uiblindmeeting/.

[29] S. Schlinker, "'Accessible Art at the Broad: Exhibit Combines Braille, Poetry and Painting." MSUToday Newspaper, 2015.