Teaching Data Structures to Students who are Blind

Matt Calder, Robert F. Cohen, Jessica Lanzoni, Neal Landry, Joelle Skaff
Department of Computer Science
UMass Boston
Boston, MA 02125
+1 617 287 6452
{mcalder, rfc, jlanzoni, nlandry, jskaff}@cs.umb.edu

ABSTRACT
We present our work in assisting students who are blind to understand fundamental data structures. We have developed a system called PLUMB EXTRA[3] (EXploring data sTRuctures using Audible Algorithm Animation) that conveys an algorithm animation using audio cues and synthesized speech. This extends our earlier work on presenting graphs to users who are blind.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Auditory (non-speech) feedback

General Terms
Human Factors.

Keywords
Accessibility, Algorithm Animation, Audio Display, Data Structures.

1. INTRODUCTION
We present our work in assisting students who are blind to understand fundamental data structures and algorithms. We have developed a system called PLUMB EXTRA[3] that conveys an algorithm animation using audio cues and synthesized speech. This extends our earlier work on presenting graphs to users who are blind [4][5][6].

Our work is motivated by our experience that so many fundamental computer science concepts are taught by first presenting an example through a diagram or animation and then moving to an algorithm and code. These visual examples are not beneficial to blind students unless they can be understood using an alternative modality, such as touch or sound. As a result of the highly visual nature of communication in disciplines such as computer science, mathematics and engineering, students with visual impairments often struggle to perform at the same level as their sighted peers. In some cases, blind students are entirely deterred by such areas of study. Therefore, our challenge is to find a way to assist a user who is blind to understand fundamental algorithms and data structures.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ITiCSE'07, June 23–27, 2007, Dundee, Scotland, United Kingdom.
Copyright 2007 ACM 978-1-59593-610-3/07/0006...$5.00.

Algorithm animations can be a constructive tool in data structure education to help students visualize the behavior of an algorithm [14]. When a student who is sighted views an algorithm animation, the representation of the data structure helps them to gain an understanding of how the steps of the algorithm change the state of the data structure. Since most educational animations run on small examples, the student gets a holistic view of effect of the algorithm on the data structure.

Our technique presents an algorithm to a student who is blind in the following manner:

- The user can incrementally move forward and backwards through the steps of an algorithm. The system uses synthesized speech to speak a brief description of each step
- At each step of the algorithm, the user can explore the state of the data structure to understand changes. We do this using our techniques presented in [5][6] and summarized below.

Beyond computer science education, our work may also accommodate students who are not visual learners. When developing novel educational tools or software, universal accessibility must be considered; PLUMB EXTRA[3] can be used to support multiple learning styles in addition to teaching students both with and without disabilities. The U.S. Individuals with Disabilities Education Act (IDEA) mandates that equal educational opportunities be provided for students in the most integrated setting possible [10]. In academia, assistive technology, tools and software such as PLUMB EXTRA[3], profoundly facilitate full inclusion in the classroom. Newell [11] emphasizes the necessity of establishing a methodology of working with users with disabilities. Some considerations include but are not limited to: user characteristics and functionality, scenarios when a universal design may not be appropriate, and provisions for accessibility with the use of additional components.

Using C#, we have implemented PLUMB EXTRA[3] in the windows environment. The application runs on standard hardware. We are currently negotiating an open source distribution license for the software.

1.1 Previous Work
For an unsighted person, a diagram can be printed as a tactile picture using a special printer. The produced diagrams are rigid and can be difficult to work with. To produce an algorithm animation using the tactile method, each step of the algorithm would be printed as a separate diagram. The user would be faced with the challenge of sequencing and matching up equivalent items of the diagram. Often, the specialized printers are costly

المنارة للاستشارات
www.manaraa.com

and difficult to obtain. Francioni and Smith [8] display tactile images using layered sequences of diagrams. This helps a student depict individual segments of the diagram while gaining an intuitive understanding of the whole diagram. Another vehicle for conveying diagrams and relational information is verbal transcription; however, an instructor may not always be able to utilize this method. Verbal transcription is appropriate only for the simplest of diagrams.

Screen readers such as JAWS [9] and Window-Eyes [10] assist blind computer users with reading documents and user interfaces. Ramen illustrates a speech-enabling approach to developing programs that separate computation and information from the actual user interface [16]. With a screen reader, blind computer users can accomplish many of the same tasks as their sighted counterparts; however, screen readers can not be relied upon for conveying otherwise visual material such as diagrams and graphs. Screen readers provide information about graphs or diagrams only through the use of alternative text, or, ALT text; a written description of the diagrams. Details are minimal and fail to give someone who is blind a conceptual and relational comprehension of the diagram. A computer science student who is blind can obtain a PDF file of a textbook from the publisher, but has access to the text using only a screen reader.

Our approach is to use audio feedback to present a diagram to a blind user. Much work has been done in sonification of relational data (see, for example, [2]]). We categorize previous systems as using either passive or active exploration. With passive exploration, the user is presented with a representation of the entire diagram at one time, with limited user input (e.g. AudioGraf [12] and the case tool of Bleckhorn and Evans [1]). With active exploration systems, the user can explore the diagram (e.g. TeDub [17] for exploring UML diagrams and Kekulé [3] for exploring molecular structures).

2. PLUMB EXTRA[3] Design and Features

This study is part of our on-going work developing novel techniques to display diagrams, graphs, and relational information to users who are blind. This work extends our software system called *exPLoring graphs at UMB* (PLUMB) [4][5][6] that displays a drawn graph and uses auditory cues to help a blind user traverse the graph.

In PLUMB EXTRA[3], the following tasks are implemented as audio through interfacing with a screen reader as well as through native speech and audio feedback (see Figure 1):

- The user selects a data structure to animate.
- The user selects a data structure operation.
- The user steps through the algorithm for the operation and observes how each step affects the data structure.

The interface is divided into 3 sections:

- A list of data structures that can be accessed from the File menu.
- A list of operations that can be performed on the selected data structure which is populated in the drop down list once a data structure is selected.
- The display, which provides both a graphical interface for sighted users and the PLUMB interfaces for blind users.

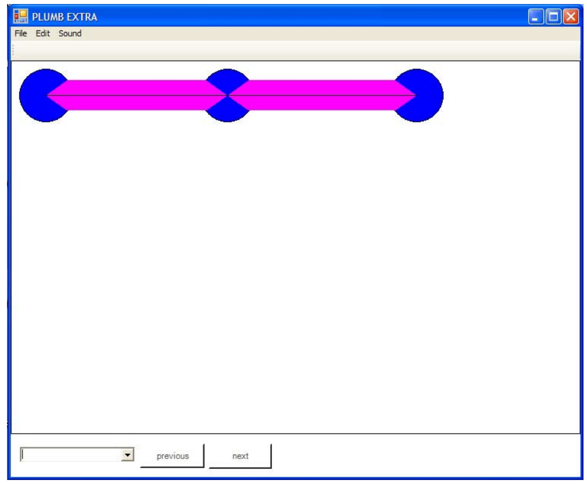

Figure 1 The PLUMB EXTRA[3] interface.

The user selects the data structure they wish to work with by hitting Alt-F on the keyboard. They are then prompted with a list of data structures to explore. As the user scrolls down, the application will announce which data structure selected via synthesized speech.

After a data structure is selected, the user then chooses an operation to animate from a list of appropriate data structure operations. As with the data structure selection, information regarding each operation is announced to the user via native speech, as well as presented graphically for sighted users.

We convey the algorithm to a student who is blind with the following interactions:

- The student selects the next and previous operations through the use of keyboard shortcuts. Descriptive text associated with each step is read to the student providing an insightful explanation of the effect of the step.
- The PLUMB interface (described above) is used to understand the state of the data structure after the step executes.

The user can run the algorithm forwards and backwards and explore the state changes. In this way they can gain an understanding of the behavior of the algorithm.

PLUMB EXTRA was designed specifically with extensibility in mind. The PLUMB EXTRA engine uses well defined software interfaces for describing how a data structure will be described by the system and what type of operations will be given to the user for modifying the data structure. These interfaces allow developers to easily add their own data structures and operations or to extend existing ones. PLUMB EXTRA uses GXL XML files to read in and save instances of an existing data structure. This portable format allows users to create and modify data structures in a variety of different editors.

2.1 PLUMB Interface

We use the PLUMB interface for exploring the state of the data structure during each step of the algorithm; Users explore the diagram and obtain audio feedback depicting elements they explore. A sample of the PLUMB visual display is shown in

www.manaraa.com

Error! Reference source not found., with the visible graph components corresponding to the locations of the underlying sound components. A representation of the same graph with associated text fields is shown in Figure 3.

Users can interact with PLUMB in two distinct modes: *keyboard mode* and *pen mode*. In keyboard mode, the user uses keyboard shortcuts to explore the graph. At any time during exploration, there is one vertex and one edge in focus. The user has keyboard shortcuts ("v" and "e") that cause the system to speak more detailed information about the current vertex and edge. Arrow keys allow the user to move around the adjacency list of the current vertex to find an edge to follow. The user hits the enter key to follow an edge and make the opposite vertex the in focus vertex (see **Error! Reference source not found.**).

Keyboard mode allows the user to quickly explore the graph and understand relationships. However, no information is conveyed about the shape of the layout of the graph. This sort of information is important when the position of the vertices and edges have some meaning in the application, such as with geographic information.

Pen mode, which requires a tablet PC, allows the user to explore a graph using the tablet PC stylus. The application generates an appropriate audio cue when the user moves the stylus over an element of the graph. We use a continuous musical tone to indicate contact with an edge, and modify the tone with a vibrato effect to indicate proximity to either end of the edge, increasing the intensity of the vibrato as the user approaches a vertex. To help users follow the edges, variations in the loudness of the musical tone are used to represent distance from the central axis of an edge, and are here depicted by variation in saturation.

In pen mode, the user "draws" on the tablet screen with the stylus to receive audio feedback and explore the graph. When the stylus enters or exits a vertex, a sound is played to communicate the event.

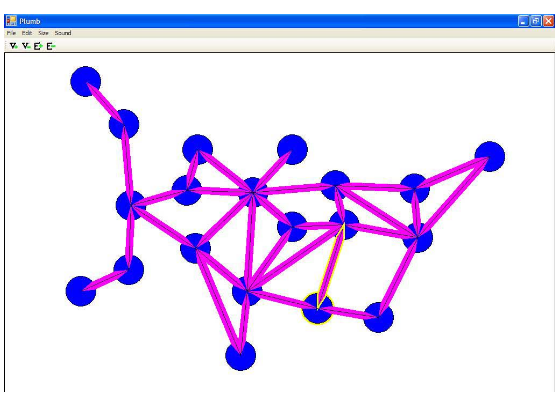

Figure 2 A screenshot of PLUMB. The displayed graph corresponds to the Europe map shown in Figure 3. The highlighted vertex and edge are in focus. The user presses "v" or "e" to hear information about the vertex or edge.

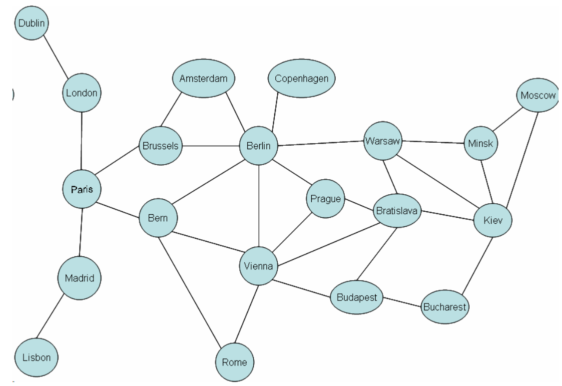

Figure 3 The European city graph displayed by PLUMB in Error! Reference source not found.**.**

Keyboard mode is sufficient for exploring most data structures in a typical data structures course. Although not part of this study, pen mode may be appropriate for animating more complex algorithms in fields such as geographic information systems and computational geometry.

3. Sample Animation

We have implemented several algorithm animations on PLUMB EXTRA³, including operations on linked lists and heaps. To make our technique clear, we demonstrate how PLUMB EXTRA³ displays an animation for inserting a node into a linked list.

When the user selects the linked list data structure, the system generates an empty linked list with dummy first and last nodes (see Figure 4). The user hears from the system "Empty linked list with first and last nodes". The user can explore the diagram and determine that there is a node called "First" with a next pointer to "Last", and a node called "Last" with a next pointer to "First". Note that the visual for the diagram is not meant to be fully comprehensible to the sighted user. It simply represents where the diagram can be found on the display (for pen-mode). The full description is found by exploring the diagram using the PLUMB interfaces.

Figure 4 The PLUMB EXTRA³ representation of an empty linked list.

When the user selects the "Insert a node" operation, the animation begins. For the first step of the algorithm, the system says "Create a new node", and the visual representation looks like Figure 5, and the user can discover that there is a new node in the diagram. Similarly, the next two steps, "set the new node's next pointer to be the same as First's next pointer" and "set First's next pointer to be the new node" are shown in Figure 6 and Figure 7.

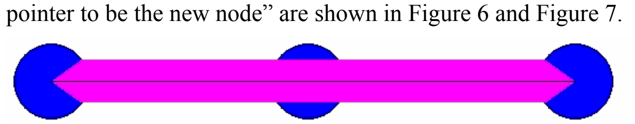

Figure 5 The first step of the algorithm for adding a node to a linked list.

www.manaraa.com

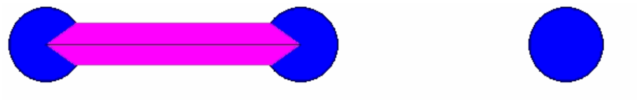

Figure 6 The second step of the algorithm for adding a node to a linked list.

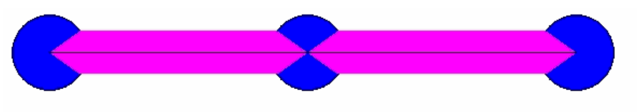

Figure 7 The final step of the algorithm for adding a node to a linked list.

4. Preliminary Results and Future Work

Although we have not completed formal trials, we did have a senior Computer Science student who is blind evaluate the system. The student has experience using the original PLUMB interface, and a deep understanding of fundamental data structures.

After a brief introduction of PLUMB EXTRA³, we had the student explore the animation for insertion into a linked list. The student added several nodes and ran the animation forward and backward.

The student successfully completed the animation with minimal prompting from the developers. The student felt that the model was natural to use. He encouraged us to continue testing the tool.

It is our hope that we have a novel, viable tool which can be implemented in the classroom. We are developing a portfolio of animations which will support each algorithm in a data structures curriculum and conduct classroom trial with blind computer science students. Trials using our PLUMB EXTRA³ curriculum will help us to measure the educational effectiveness of the tool. We hope to move beyond basic data structures and use this tool for more complex algorithms and data structures.

5. ACKNOWLEDGMENTS

We would like to thank Nancy Lightbody and the Eastern Alliance in Science, Technology, Engineering, and Mathematics (EAST) for supporting us in this work. We would also like to thank David Gerry and Valerie Haven for useful discussions and feedback that helped us in designing the system.

6. REFERENCES

[1] Blenkhorn P., Evan D. G. Using Speech and Touch to Enable Blind People to Access Schematic Diagrams. Academic Press, 1998, vol. 21, no. 1, pp. 17-29.

[2] Brewster, S.A. Theses: Providing a structured method for integrating non-speech audio into human-computer interfaces. 1994. Available at http://www.dcs.gla.ac.uk /%7Estephen/publications.shtml#4.

[3] Brown A.,Pettifer S.,Stevens R., Evaluation of a Non-Visual Molecular Browser. In Proceedings of ASSETS'04 (Atlanta, GA USA, October 18-20, 2004), 2004, pp. 40-47.

[4] R.F. Cohen, V. Haven, J.A. Lanzoni, A. Meacham, J. Skaff, and M. Wissell. "Using an Audio Interface to Assist Users Who are Visually Impaired with Steering Tasks," *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS'06), Pages 119-124 (2006).

[5] R.F. Cohen, A. Meacham, and J. Skaff. Teaching Graphs to Visually Impaired Students Using an Active Auditory Interface, *Proceedings of the 2006 Technical Symposium on Computer Science Education* (SIGCSE 2006), pages 279-282 (2006).

[6] R.F. Cohen, R. Yu, A. Meacham, and J. Skaff., PLUMB: Displaying Graphs to the Blind Using an Active Auditory Interface, Proceedings *of the 7th international ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS'05), pages 182-183 (2005).

[7] *Cornell News*, Available at http://www.news.cornell. edu/releases/Jan05/Wong.software.to.html

[8] Francioni, J. M. and Smith, A. C. 2002. Computer science accessibility for students with visual disabilities. *SIGCSE Bull.* 34, 1 (Mar. 2002), 91-95.

[9] Available from Available from Freedom Scientific, http://www.freedomscientific.com

[10] Available from GWMicro, http://www.gwmicro.com/.

[11] Holt R. C., Winter A., Schürr A., GXL: Towards a Standard Exchange Format. In *Proceedings 7th Working Conference on Reverse Engineering (WCRE 2000). Science of Computer Programming*, Elsevier Science, 2005, pp. 34. Available at http://www.gupro.de/GXL/.

[12] Kennel A.R**.** Audiograf: a diagram-reader for the blind. In *Proceedings of the second annual ACM conference on Assistive technologies* (Vancouver, British Columbia, Canada). ACM Press, New York, NY, 1996, 51-56.

[13] Slatin J., Rush S. 2003. *Maximum Accessibility.* Boston: Addison-Wesley.

[14] Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., and Velázquez-Iturbide, J. Á. 2003. Exploring the role of visualization and engagement in computer science education. *SIGCSE Bull.* 35, 2 (Jun. 2003), 131-152.

[15] Newell, A. F. and Gregor, P. 2000. "User sensitive inclusive design"— in search of a new paradigm. In *Proceedings on the 2000 Conference on Universal Usability* (Arlington, Virginia, United States, November 16 - 17, 2000). CUU '00. ACM Press, New York, NY, 39-44.

[16] Raman, T. V. 1997. *Auditory User Interfaces: Toward the Speaking Computer*. Boston: Kluwer Academic Publishers.

[17] The TeDUB-Project (Technical Drawings Understanding for the Blind). Available at http://www.kinf.wiai.uni-bamberg.de/research/projects/TeBUB.

www.manaraa.com