DEALING WITH UNSEEN OBSTACLES TO EDUCATION IN THE DIGITAL AGE

Valerie J. H. Powell[1], Arif Sirinterlikci[2], Christopher Zomp[1], Randall S. Johnson[3], Phillip Miller[3] and James C. Powell[4]
[1]Robert Morris University Department of Computer and Information Systems
[2]Robert Morris University Department of Engineering
[3]Robert Morris University IT Services, [4]Better Schooling Systems

ABSTRACT

This paper updates the efforts to educate blind students in higher education in the digital age and describes how to support the development of mental models in learning through tactile learning and 3D-printing technology. It cites research documenting a drop in Braille literacy along with the growth in use of digital technologies by blind students. It identifies technologies, strategies, and techniques used to address barriers to progress faced by a student majoring in computer and information systems, where learning to program a computer and design databases were requirements for a degree.

KEYWORDS

Blind, computer programming, mental model, 3D printing

1. INTRODUCTION

During the digital age literary Braille usage has been dropping and fresh barriers for visually impaired learners are posed through online education and new visually-oriented web page interfaces (Thompkins, 2010; Perry, 2011; Avin, 2010). Novel digital technologies provide access to text in audio format, such as screen readers, ebooks, and mp3s (Avin, 2010). We focus here on the needs of the severely visually impaired (SVI) student and on the teaching/learning process, according to the Universal Design of Learning (UDL) (CAST, 2011) first principle of providing “multiple means of representation.”

The SVI student must transcend sole reliance on audio inputs to confirm concepts of structure and relationships vital in learning about information technology. After a model (hypothesis) of a structural relationship is formed, it “can then be tested by the agreement or disagreement of predictions based on it with new facts which are discovered and its merits assessed by the degree of modification which it requires in order to meet these new facts.” Craik (1943). How should a relationship be modeled for a blind learner? Abidin et al (2012, 2013) explored blind persons' mental models of web pages and investigated the relationship between users' mental models and usability problems they face when using a screen reader and found that blind users possess a functional or structural mental model or a combination thereof.

2. INSTRUCTIONAL ENVIRONMENTS

Here are the instructional environments used for teaching the SVI student at Robert Morris University (RMU), with their advantages, disadvantages, and special provisions or accommodations:

2.1 Ordinary Classroom Setting

The *advantages* are that (a) the student is integrated with other (sighted) students, as he or she would be in a work situation, (b) the student receives all the *auditory information* that all students in the group receive.

The *disadvantages* of this setting are that any diagram on the board, drawn by the instructor or other students, or distributed to all students on a sheet of paper, are useless for the SVI student. Sighted students in the classroom may become impatient for attention if attention to the SVI student's needs requires more than a few seconds. *Accommodations* are to arrange for a sighted student sitting next to the SVI student to interpret situations and explain diagrams or work on the board. Printed or posted instructions that are adequate for sighted students are often inadequate for an SVI student, especially where the student must learn to use a new computing environment. Occasional one-on-one meetings (perhaps 1 per week) can be scheduled to assure that the SVI student has adequate opportunity to ask questions and have them satisfactorily answered in detail. These occasional meetings offered an opportunity for assessment through dialog to help the instructor track the SVI student's progress and to verify that all instructions on how to use a computing environment are usable. Relying solely on class meetings and student questions during class is likely to be inadequate.

2.2 One-on-one Classroom Setting

The *advantages* are that (a) the student receives individual attention and support. The teacher doesn't have to worry about pacing impacts on other students. (b) The instructor has the opportunity to maintain running class notes for both to consult subsequently. Sometimes the student needs time to refine a screen navigation strategy which can then be documented in the Class Notes. (c) student and instructor discuss strategies as needed to solve technical challenges. *Disadvantages* may be that (a) the instructor may not be patient enough for the SVI student and may prompt him/her on the basis of what the instructor can see when the student should be allowed to work through the situation at his/her own pace and learn to detect problems. *Accommodations* include class notes, joint instructor/student evaluation of technology options, such as screen readers.

2.3 Over-the-phone Setting

Where both student and instructor have access to a multiuser server, an over-the-phone instructional setting can be practical in exploring the learning of programming. The *advantages* are, ironically, that the instructor cannot see the SVI student's screen and is not physically present and thus cannot prompt the student based on what the instructor can see. The student must be self-sufficient. *Disadvantages* are that the instructor cannot see the SVI student's screen and is not physically present and thus cannot help resolve certain problems that may arise and may even lead to termination of the session. *Accommodations* include multiuser access to systems, and two models of this were employed: (a) multiuser access to an InterSystems Caché ObjectScript server and (b) access to two Linux virtual machines in a virtual machine network.

3. SUPPORT TECHNOLOGIES AND PROCEDURES

This section describes the learning support strategies adopted in teaching INFS3141 (Advanced M Programming) and INFS4953 (Intrusion Detection). When the student chose to major in Information Systems, he faced a requirement for completing two courses in programming in the same language.

Eventually he decided on M programming from the options available. What were the advantages and disadvantages of using the InterSystems Caché ObjectScript application programming interface (API)? Prior to working with InterSystems Caché, our SVI student had been counseled to try COBOL. He found this not practical. Caché (M Programming) was then recommended as the programming can be carried out with single-letter commands and single-letter or two-letter function identifiers and the programs are not as verbose as those coded in other computer programming languages. While database capabilities are integrated with the programming language, the database does not have to be declared, as is the case with SQL. M databases are considered to be NOSQL databases and can support the largest datasets. The industries in which M programming is used include, in the U.S., health care and the financial industry (banks and credit unions, stock market, and even space research (InterSystems, 2012)). One disadvantage is that the Caché API Studio has an editing error indicator which, so far, is not accessible to SVI learners.

3.1 Screen Reader Selection and Utilization

Screen-reading technologies, predominantly JAWS (2013), were employed by the student in this project. His experience with Braille had been unsatisfactory so he had meager Braille capabilities. JAWS can distinguish between *upper and lower case* by voice pitch. Because of the significance of case sensitivity in programming environments, this is an important capability of JAWS. Experimentation showed that no single screen reader met his needs in learning programming. JAWS sometimes does not render all program output on the screen, dropping whole sections of software output. This led to the student using a second reader, NVDA (2013), which added to the complexity of managing these technologies and comprehending and verifying program output, with two screen readers sometimes generating spoken text at the same time.

3.2 Tactile Access to Diagrams

A major challenge in educating an SVI student is the prevalence of visual materials in curriculum. Diagrams and illustrations are common, delivered ordinarily to students through software, instructional management systems like Blackboard, Power Point presentations, through faculty and student drawings on classroom chalkboards and whiteboards, graphic material in textbooks and Internet links, and as an object for discussions. Eventually it was apparent that the most significant barriers learning were posed by certain key conceptual diagrams employed on a variety of courses. Examples are:

Table 1. Instructional Diagram and Graphic Image Examples
Entity-relationship (ER) diagrams in database design
Database Relationship Diagram as displayed by Microsoft Access (See Figure 1)
Unified Modeling Language (UML); tactile access documented by Brookshire (2006).
Relational database design: table relationships and cardinalities (SQL standard is ISO/IEC 9075-1:2011; see Figure 1)
Models of Caché ObjectScript global (or local) variables as databases (standard: ISO 11756; see Figure 2)
Comparisons of the relational and hierarchical database models in courses in which the student enrolled (INFS3141 and INFS3450)
Maps; accessibility strategies documented by Koch (2012).

Following UDL principle I led to consideration of audio and tactile feedback to deliver multiple representations. Audio feedback for diagrams or graphs was covered by Cohen et al (2006), Sánchez (2008), Kennel (1996), Sánchez and Flores (2010), Abidin et al. (2012, 2013). Tactile feedback was covered by Petrie et al. (1995), Brookshire (2006), Lévesque (2008), Koch (2012). Textures in tactile models was covered by Abidin et al. (2012). Haptics was covered by Lévesque. The decision was made to pursue static tactile models for diagrams.

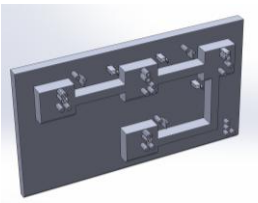

Figure 1. Relational Database Model

Figure 2. M Array (Global) Database Model

The design of 3D tactile models had to consider whether Braille numbers should be displayed on the model (Figure 1). The student did not commonly use Braille but could read Braille numbers. Number symbols are required before each numeral in an environment where both numeric and nonnumeric symbols are used.

Commonly used Braille did not have a symbol for infinity, so we either had to use a different Braille standard or use the letter “m” for *many* in expressing database diagram cardinality. Brookshire (2006) solved this problem simply for his tactile models. For the second model (Figure 2), Braille was not used.

Both student and instructor assessments and concerns were required to identify the models or diagrams of greatest need and thus prioritize the process of producing 3D models.

Input from the student on 3D images began with the relational database model shown in Figure 1. The student characterized use of the 3D model of a diagram as a novel, unprecedented experience and reported that it helped him gain an "accurate interpretation" and that he revised his mental model of database relationships based on his experience with the 3D model. The Braille numbers on the original model was discovered to be superfluous because (1) the student could interpret the Arabic number representations as effectively as those in Braille and (2) mixing numeric and alphabetic symbols required more symbols, and (3) ordinary Braille was not designed to present scientific and technical symbols only available in specialized versions of Braille, such as Nemeth Braille (AAWB, 1987).

Learner input regarding the M array model (Figure 2) revealed that he gained new insights from using the model. The student practiced two database array traversal functions of Caché ($data() and $order()), while consulting the model through touch. $Query() function array traversal was also aided through use of the 3-D model.

The models presented in Figures 1 and 2 were designed in Solid Works and printed using the Fused Deposition Modeling (FDM) process by employing a Dimension Elite Printer and ABS (acrylonitrile butadiene styrene) industrial plastic. No chemicals such as sodium hydroxide were utilized in removing the support materials, but mechanical loading was used to prevent contamination of the models.

Front edges of raised blocks as well as the base plate for both designs were chamfered and the lower left corner of the design illustrated in Figure 1 had a number sign in Literary Braille to establish orientation for the SVI user. Design illustrated in Figure 1 had also Nemeth Braille Symbols representing infinity with a comma and an equal sign while rest of the raised information was given in either Numbers or Literary Braille. The design illustrated in Figure 2 was printed in two pieces and assembled through a snap fit connection.

Even though current FDM models are not cost effective, they can be improved with use of less material, also by lowering process times and costs which is more critical than material costs. Reductions in process and materials costs due to the new concept of Rep-Rap (Replicating Rapid Prototypers) will make printing 3D tactile models economically justifiable.

Design of inexpensive refreshable (programmable) diagrams may pose another alternative along with 3D printed tactile models. These flexible programmable tools can hold a major advantage over their 3D printed counterparts with the addition of voice feedback coupled with their tactile content.

Assessment of the impact of the 3D model of the M array consisted of exercises involving the identical array as modeled and of additional array models, in order to discover what learning, if any, had occurred through consulting the 3D model. It was apparent that the learner had achieved a generalized model valuable for approaching arbitrary alternative array instances.

3.3 Multiuser Server and Virtual Machine

RMU utilized Amazon Web Services and built an EC2 server instance for this project. InterSystems Caché for Windows (x86-64) 2012.1 (Build 564U) was installed this Windows 2008 R2 server. Caché user accounts were created for students and faculty. While Caché can be installed locally for development and testing purposes, this server-based multi-user environment allowed faculty and students a real-world experience with shared database objects, scripts, and data.

As an example of software used by an SVI student in the multiuser server environment is the `retail4/retailm` combination. These Caché ObjectScript routines are designed to demonstrate and give practice with the following concepts: multiuser access to a common central database by different kinds of processes, designing and using a database to document transaction history, generating unique numbers for transactions, object locking and concurrency control. With access to the RMU Caché ObjectScript multiuser server, student and faculty can invoke the `retail4` or other routines remotely from different locations while conversing on the phone.

3.4 Class Notes

The development of "class notes" for face-to-face meetings was employed in teaching the SVI student. Considering the extent of detail in a programming course it seemed prudent to develop a record of transactions in a class meeting to give the student detail on what was covered and to assure a way to review details prior to a job interview or during employment in the field. It is important to capture the student's problem-solving achievements in the Class Notes. The disadvantage of the Class Notes as carried out is likely their uneven quality and lack of consistent organization or of indexing.

Table 2. Proposed Objectives for Class Notes
Document challenges and technical problems encountered and describe techniques for solving technical problems, especially of managing screen readers and navigating the API and the screen in general.
Identify and document strategies, habits, and procedures for quality control, such as assuring that potential error indications have been routinely checked, making certain where the cursor is on the screen before proceeding, verifying program operation, etc.
Identify and document programming challenges, such as left and right quote marks in code text copied from word-processing of web-based sources.
Document application programming interface (API) limitations discovered as a basis for vendor followup.
Document relevant and informative links discovered.
Record interesting and useful examples of code.

Preparing printed materials for the SVI student required thought and negotiation to assure ease of use. Editing and control interfaces encountered in this project were InterSystems Caché Studio and a Linux editor, such as *vi*, and WinSPC. Instruction should cover the pitfalls of copying and inserting text.

4. LIMITATION, CONCLUSION, AND RECOMMENDATIONS

The limitation of this study that it is based only on a single set of experiences with a single student must be remedied by collaborating with programs at other universities facing similar needs. A plan to develop a broader project to obtain more general results has been devised.

The pervasive use of visually-oriented instructional approaches in web pages and on-line instruction and the use of virtual machine software to deliver secured access to all students may ignore the accessibility requirements of SVI learners. A systematic approach to SVI accessibility is required to assure opportunity for learning in the digital age. Multiple representation modes, as recommended by UDL, is a vital strategy to improve accessibility, as the student can then consult feedback from a second representation mode to assess the adequacy of an initial mental model. Today's SVI students rely less on Braille and more on screen readers. These students are familiar with and customarily use electronic communication technologies. Specific recommendations follow in Table 3.

Table 3. Recommendations
Assess the individual SVI student's competence with literacy (e.g. Braille) and accessibility tools (e.g. screen readers) upon admission to <university/college>. Take assessment results into account to plan accommodations for the student. Support and enhance the student's preferred and accustomed modes of access to instructional materials. Have support guidance ready and available for each SVI student and instructors. Faculty should receive guidance information before the term begins to the extent feasible.
Decide on appropriate faculty development to make available for instructors of each SVI student and for all interested faculty. Consider including an orientation to SVI accessibility in a faculty meeting.
Develop guidance for teaching SVI students with lists of resources. Ideas might be gained from the page posted by the University of Northern Iowa: http://www.uni.edu/walsh/blindresources.html or from Pennsylvania State University's page: http://accessibility.psu.edu/visuallyimpaired. Encourage accessibility by involving any given student in the evaluation of accessibility. Recommend an approach to teaching that incorporates periodic assessment during the course term of accessibility of course instructional materials. Require every instructor of an SVI student to consult guidance and sign off.
Review accessibility of online courses, course web pages, and instructional support software (e.g. VMWare) to assure compliance with ADA. Involve the local Center for Educational Technology and IT Services. Assure that information

and guidance is provided to each online course developer. Use the University of Illinois Function Accessibility Evaluator 1.1 (FAE): http://fae.cita.illinois.edu/ and consult the resource "Creating Accessible Web Pages" from the University of Colorado: http://www.colorado.edu/webcom/access/, W3C's "Web Content Accessibility Guidelines" (2008) and other such resources as needed. Act in accordance with findings to ensure accessibility.
Review options for providing audio and/or haptic/tactile diagram or image accessibility. Examples of diagrams: Unified Modeling Language (UML), Entity Relationship Diagrams (ERDs), M Programming Global Arrays. Examples of image graphics: maps, Geographic Information System (GIS) outputs. Develop procedures and budget for making accessibility support available for SVI students.
Develop procedures for recording class notes for SVI students if desired. Ascertain any given student's preferences for such note-taking. Determine if any given SVI student wishes a "header" symbol for indexing documentation and, if so, which one.

REFERENCES

Avin, Rachel (2010) "Listening to Braille," *New York Times* January 3, 2010, page MM42. http://www.nytimes.com/2010/01/03/magazine/03Braille-t.html?_r=0. Accessed April 22, 2013.

Abidin, Ahmad Hisham Zainal (2013), correspondence, April 24.

Abidin, Ahmad Hisham Zainal, Xie, Hong, Wong, KokWai (2012). *Blind Users' Mental Models of Web Page Using Touch Screen Augmented With Audio Feedback*. 2012 International Conference on Computer and Information Science (CCIS). Accessed April 22, 2013.

American Association of Workers for the Blind (AAWB), Association for Education of the Visually Handicapped, and the National Braille Association (1987). *The Nemeth Braille Code for Mathematics and Science Notation*, 1972 Revision (American Printing House for the Blind). https://nfb.org/images/nfb/documents/pdf/nemeth_1972.pdf Accessed June 2, 2013.

Brookshire, Robert G (2006). "Teaching UML Database to Visually Impaired Students." *Issues in Information Systems* 7 (1), 98-101.

CAST (2011). Universal Design for Learning (UDL) Guidelines: Full Text Representation.. http://www.udlcenter.org/aboutudl/udlguidelines

Cohen, Robert F, Meacham, Arthur, Skaff, Joelie (2006). "Teaching Graphs to Visually Impaired Students Using an Active Auditory Interface." *SIGCSE '06* (ACM), pp. 279-282.

Craik, Kenneth (1943). *The Nature of Explanation*. (Cambridge).

Freedom Scientific (2013). "JAWS." http://www.freedomscientific.com/products/fs/jaws-product-page.asp Accessed April 20, 2013.

InterSystems (2012). "European Space Agency Chooses InterSystems Caché Database For Gaia Mission to Map Milky Way" http://www.intersystems.com/casestudies/cache/esa.html Accessed April 19, 2013

Kennel, Andrea (1996). "AudioGraf: A Diagram-Reader for the Blind." *Assets '96 Proceedings of the second annual ACM conference on Assistive technologies* (ACM), pp. 51-56

Koch, Wolf Günther (2012). "State of the Art of Tactile Maps for Visually Impaired People." *Lecture Notes in Geoinformation and Cartography*, (Springer), pp 137-151.

Lévesque, Vincent (2008). "Blindness, Technology and Haptics" Accessed at http://www.cim.mcgill.ca/~haptic/pub/VL-CIM-TR-05.pdf on April 20, 2013.

Perry, Marc (2010). "Colleges Lock Out Blind Students Online," *Chronicle of Higher Education*, December 12, 2010. Accessed at http://chronicle.com/article/Blind-Students-Demand-Access/125695/ on April 21, 2013.

National Federation for the Blind (1996). "Braille: What is it and what does it mean to the blind?" *Future Reflections* 15, 1. https://nfb.org/images/nfb/publications/fr/fr15/issue1/f150113.html. Accessed April 19, 2013.

National Federation for the Blind (n.d.). "Braille Usage." https://nfb.org/braille-usage-chap1

Non-Visual Desktop Access Community (2013). "NVDA." http://community.nvda-project.org/ , Accessed April 21, 2013

Pennsylvania State University, "Accessibility and Usability at Penn State" http://accessibility.psu.edu/visuallyimpaired. Accessed April 19, 2013

Sánchez, Jaime, Flores, Héctor (2010). "Concept Mapping for Virtual Rehabilitation and Training of the Blind," *IEEE Transactions on Neural Systems and Rehabilitation* 18(2), 210-219. http://www.captura.uchile.cl/handle/2250/11226 Accessed April 22, 2013.

Sánchez, Jaime (2008). "User-Centered Technologies for Blind Children," *Human Technology* 4(2), 96–122.

Thompkins, Al (2010). "Decline in Braille Use threatens Literacy." Poynter.org. http://www.poynter.org/latest-news/als-morning-meeting/100127/decline-in-braille-use-threatens-literacy/ Accessed April 21, 2013.

World Wide Web Consortium (W3C; 2008). "Web Content Accessibility Guidelines" http://www.w3.org/TR/WCAG20/ Accessed April 21, 2013.