Circumventing Graphical User Interfaces in Chemical Engineering Plant Design

NOEL ROMEY
Ralph E. Martin Department of Chemical Engineering
University of Arkansas
Fayetteville, AR

RACHEL M. SCHWARTZ
Department of Psychology
University of Arkansas
Fayetteville, AR

DOUGLAS BEHREND
Department of Psychology
University of Arkansas
Fayetteville, AR

PETER MIAO
Chemstations, Inc.
Houston, TX

H. MICHAEL CHEUNG
Department of Chemical and Biomolecular Engineering
The University of Akron
Akron, OH

ROBERT BEITLE
Ralph E. Martin Department of Chemical Engineering
Department of Biological and Agricultural Engineering
University of Arkansas
Fayetteville, AR

ABSTRACT

Graphical User Interfaces (GUIs) are pervasive elements of most modern technical software and represent a convenient tool for student instruction. For example, GUIs are used for [chemical] process design software (e.g., CHEMCAD, PRO/II and ASPEN) typically encountered in the senior capstone course. Drag and drop aspects of GUIs are challenging for students with little or no visual acuity. We report on the use of several innovations to circumvent such aspects of GUIs.

Keywords: Visual imparity, plant design, chemical engineering

INTRODUCTION

ABET outcome (k), *an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice,* is an important requirement for student graduation. Germane to outcome (k) is the use of simulation software, for such software represents a key engineering tool that students must master to practice engineering. For chemical engineers, process simulation software is used to solve material and energy balances for proposed manufacturing schemes or to analyze current processes for possible improvements (i.e., debottlenecking). Examples of software include PRO/II by Simulation Sciences, Inc. (www.simsci-esscor.com), ASPEN by ASPEN Technology, Inc. (aspentech.com), and CHEMCAD by Chemstations, Inc. (http://www.chemstations.net/). Each software package functions similarly, and familiarity with one confers a transferable knowledge base to operate another provided the basic input/output (I/O) routines are understood. Choices of components and flow rates, thermodynamic methods, unit operations, and unit connectivity are provided by the user as input. The calculation engine then performs the material and energy balance, sizes or rates equipment, and in some cases provides an estimate of the capital investment.

Recent advances in simulation software design have moved from text based (keyword and programming language code) input to Graphical User Interfaces (GUIs). GUIs are exceptionally attractive to professors, because they allow for a significant time savings during student training [1, 2]. Simply put, classroom exercises are freed of debugging the keyword syntax required to run the calculation engine. Since *pointing and clicking* generates the code, presumably more time is spent understanding the chemical process being studied and how it may be correctly designed or improved. For example, the simulation of a facility to produce ethylene oxide described in the popular design textbook *Analysis, Synthesis, and Design of Chemical Processes* [3] can be built with a GUI by dragging 23 appropriate unit operation icons to a design palette, indicating connectivity with lines between icons, filling in appropriate drop down menus, and then pushing the run button. This generates approximately 200 lines of keyword code by PRO/II, for example, which is then processed through the calculation engine.

Consider, however, the difficulty associated with using a GUI if a person has limited or no vision. Classroom activities must obviously accommodate this student. Two avenues exist for student accommodation: (i) falling back, so to speak, to the exclusive use of keyword files or (ii) adapting the I/O routine of the software package. We believe that completely circumventing the use of a GUI may not be a viable long-term option for a visually impaired student. The overall requirement of providing instruction for all design students within the context of ABET outcome (k) makes the use of GUIs invaluable. However, for a federally funded university to be in compliance with section 504 of the Vocational Rehabilitations Act of 1973 (Public Law 93-112), one recognizes the

fact that accommodations must be made for all qualified students. Although the visually impaired student could literally step backward in time approximately 12 years before the advent of the GUI and work solely with code, the quality of modern instruction would be sorely compromised for several reasons:

(1) Time allotted for creative exercises would be diminished to allow for the writing and debugging of keyword files.

(2) Avoiding GUIs *conflicts* with ABET outcome (k).

(3) A separate and distinct tract for the visually impaired student only delays the inevitable.
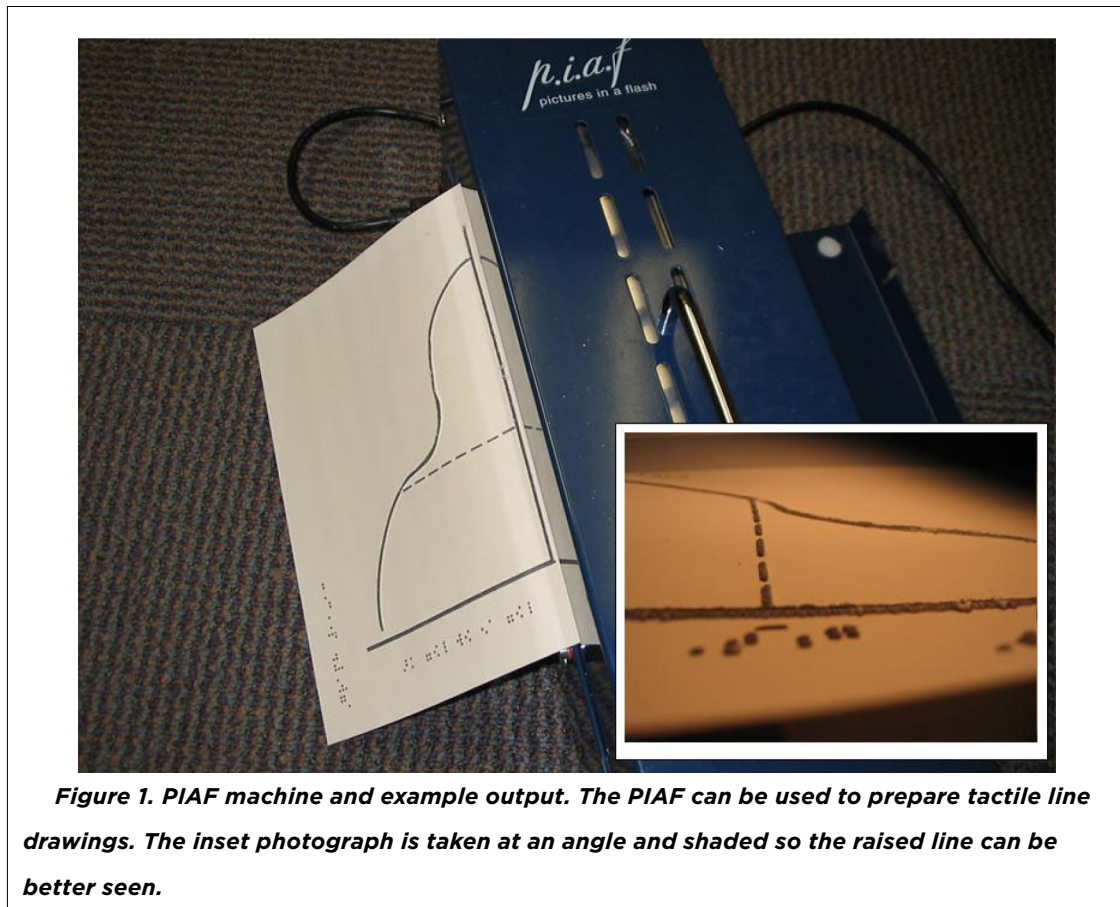
Item (3) warrants further comment. Since the visually impaired student would function in a group setting with sighted team members, a disparity would exist simply because each would approach process simulation differently. Those able to employ a GUI would arguably try more possible solutions, whereas those constrained to keyword code may examine fewer scenarios. To be cliché, *history would repeat itself:* imagine returning to a time in the classroom where one or two simulations were examined due to programming constraints. Such scenarios are clearly not acceptable in modern chemical engineering education and practice.

To this end, our group has begun work towards minimizing the differences between visually impaired and sighted students. This can be achieved through a combination of techniques that adapt I/O in a creative fashion. We report on combining screen reading programs, audible cues, and finally, tactile representations of screen icons to provide access to modern simulation software for a visually impaired chemical engineering student. To our knowledge, this multifaceted approach has not been attempted with technical software packages.

BACKGROUND

Current "One Way" Methods

There are several methods used by educators of the visually impaired to convey scientific or mathematical content. These techniques typically require some visual acuity to produce material for the student. Techniques include complicated and costly methods such as Pictures in a Flash (PIAF) or simple methods such as raised line drawing kits [4–6]. PIAF, or "toasting" as it is routinely called, is a technique that transfers material from a master to *swell paper* using heat (Figure 1). First, an image is produced from material be it hand made or graphic printout. The image is then transferred using a simple copy machine onto the *swell paper*, then raised by exposing the copy to a concentrated light source. PIAF is more costly because of the paper and light box, but can quickly prepare material for a student. At the opposite end of the cost spectrum is the raised line drawing

Figure 1. PIAF machine and example output. The PIAF can be used to prepare tactile line drawings. The inset photograph is taken at an angle and shaded so the raised line can be better seen.

kit. By hand, an image is transferred to a thin acetate film using pressure. A stylus similar to those packaged with PDAs is used to score the film, producing a line.

PIAF or similar methods are "one way" inasmuch as a person with visual acuity is required. Such methods are considered one way because they are not interactive. Rather, static representations of material like graphs or diagrams are produced for a student to interpret.

Screen Reading Programs

In this digital age it is imperative for the millions of limited visually impaired and fully blind people living around the world to have access to computers and related technology. Although digital access has evolved and developed over the years, the basic idea of using screen reading software dates back to the days of early personal computing and has remained the same. At that time, screen readers existed (TextTalker for Apple II, VocalEyes for DOS) that were able to vocalize the screen content as new information was printed to it in a line by line fashion. Since an application only had text strings and numerical values as input or output, manipulating

information was relatively easy. Though graphics did exist and limited GUIs were available, the main media by which information was conveyed was text based, due to the limited computer power that was available.

With faster processors, more memory, and cheaper storage media, GUIs became more profitable to build, maintain, and run on desktop computers. *Pointing* and *clicking* is the current standard by which I/O is achieved. Microsoft introduced the Windows Operating System, generating selection pressure towards the creation of "attractive" software relying on GUIs. Since visually impaired users were required to manipulate GUIs, utility programs were created that could coherently read graphical information on the screen. Starting in 1997, Microsoft's Active Accessibility (MSAA) team built and continue to build special tools [7], APIs (Application Programming Interfaces), and accessibility standards into the Windows OS that allow leading screen reading packages (Jaws for Windows [Freedom Scientific, Inc, http://www.freedomscientific.com/fs_products/software_jaws.asp.], WindowEyes [GwMicro, Inc., http://www.gwmicro.com/Window-Eyes/]) to provide computing accessibility to visually impaired professionals [7].

Unfortunately, the architecture of many GUI-based technical programs does not employ MSAA features. This creates a dilemma for visually impaired users. When one launches an application using non-standard controls (i.e., graphics or unlabeled objects and buttons) that the screen reader can not recognize or interpret via MSAA or through the Windows API, the blind person is plunged back into illiteracy. Such is the case with plant simulation software noted above; i.e., CHEMCAD, ASPEN, and PRO/II. Although screen readers can interpret some aspects of the interface, many times crucial parts of the GUI can not be manipulated with screen reading programs. One critical aspect of these interfaces includes connecting unit operations via *point*, *click*, and/or *drag*.

Imbedded Audio Cues

Audio cues may be imbedded in the software to mimic events normally indicated by visual cues such as cursor shape change. This strategy requires a willingness on the part of the software developer to adapt the software package. For users with only limited vision, the accurate location of screen items is difficult or may even be impossible to find. To overcome these difficulties, distinct sounds can be introduced. For example, consider when a user is trying to indicate a stream connection from Unit A to Unit B. Conceptually, the user moves the cursor around the icon of Unit A and encounters an available port. When the port is found, a sound can be played to indicate successful capture of the port at either Unit A or Unit B. Sounds may also be used to indicate other functions as well. With the feedback of sounds, users with limited vision may be promptly informed of events that screen reading programs can not indicate.

Tactile Representations

Finally, tactile representations may be used to locate and place GUI objects. Tactile representations can be as simple as a small sticker placed on a transparency or as complex as a stenciled analog of the icon being represented. A stencil may be designed to mimic the same size of the onscreen icon and provide the user with the needed level of detail to indicate connectivity. Without tactile representations it is virtually impossible to design a facility using the *drop* and *click* features of a GUI.

METHODS

In order to examine the effectiveness of combining screen reading, imbedded audio cues, and tactile representations to allow a visually impaired person to operate a GUI, we utilized CHEMCAD version 5.5 as a test case. Although one script file for JAWS was written to provide for the reading of certain popup information boxes, most of the GUI features of CHEMCAD were accessed, or initiated, via the adaptations described above. A Tablet PC (slate model, M1200) made by Motion Computing (Austin, Texas) equipped with a USB connected keyboard was used during the course of the study. The visually impaired student ($n \overset{\sim}{5} 1$), who is unable to see, used this system to design various chemical manufacturing schemes and several separation systems for a hydrocarbon mix.

DISCUSSION

As stated previously, chemical engineers use plant simulation software that has a great dependency on GUIs. Though screen readers can inform the user of certain onscreen changes, automatically move the mouse to a predefined point, and duplicate certain mouse actions, it is impossible to tell the screen reader what to do or where to *click* if there is no *a priori* spatial knowledge. Such is the case when one is tasked with the design of a chemical plant in any GUI environment. As the screen reader can not manipulate graphics alone, a method was needed to solve this problem.

At the center of the solution was the use of tactile representations layered above a hardware element be it an Iveo touchpad [8] or Tablet PC. Two options were considered because each represented different means by which the pointer is tracked, and *point* and *click* occurs. With the Iveo touchpad (Viewplus Inc.) the user interfaced with the computer via a USB connected, letter size paper touchpad (8.5" x 11"). When the user touched the touchpad, a mouse *click* occurred and was registered spatially. Outside of native Iveo applications, the touchpad did not easily allow for *drag* and *drop* or selection using a mouse *click plus sweep* of an area. These

shortcomings made design with the Iveo impossible with the native, current drivers provided with the touchpad.

Use of the Tablet PC was successful (Figure 2). The Tablet PC registered a *click*, not with a finger touch but with the use of a special pen. This allowed the visually impaired student to use his fingers to "explore" the tactile representations and connection points without spurious *clicks* of the mouse. Moreover, both a right *click* and left *click* were possible using the pen. When the pen was within approximately one cm from the Tablet screen, the pointer *hovered* over the pallet or finished design, allowing the user to perform certain actions. Specific to CHEMCAD, the user had the ability to *hover* in order to read various popup windows describing conditions such as flows, temperatures, pressures, and/or sizing of the plant components under the elevated pen point. The visually impaired student using a screen reader could easily read these popup windows much like his sighted counterparts provided the JAWS script file (available from the paper's authors) was installed.
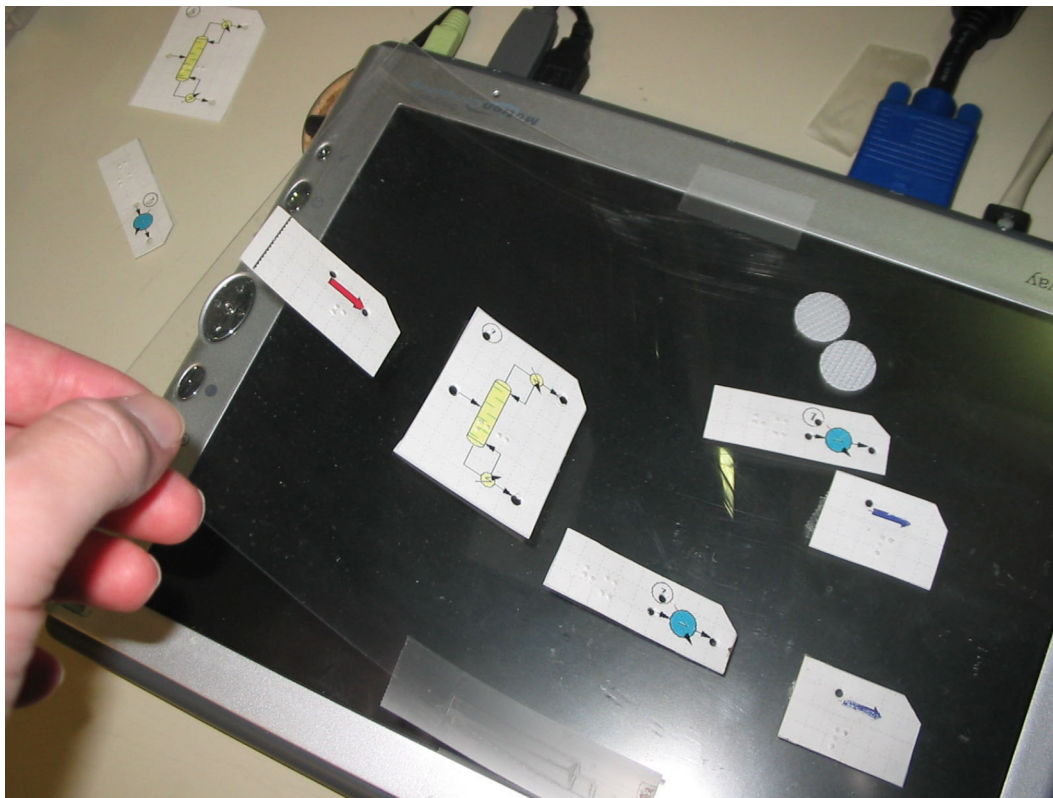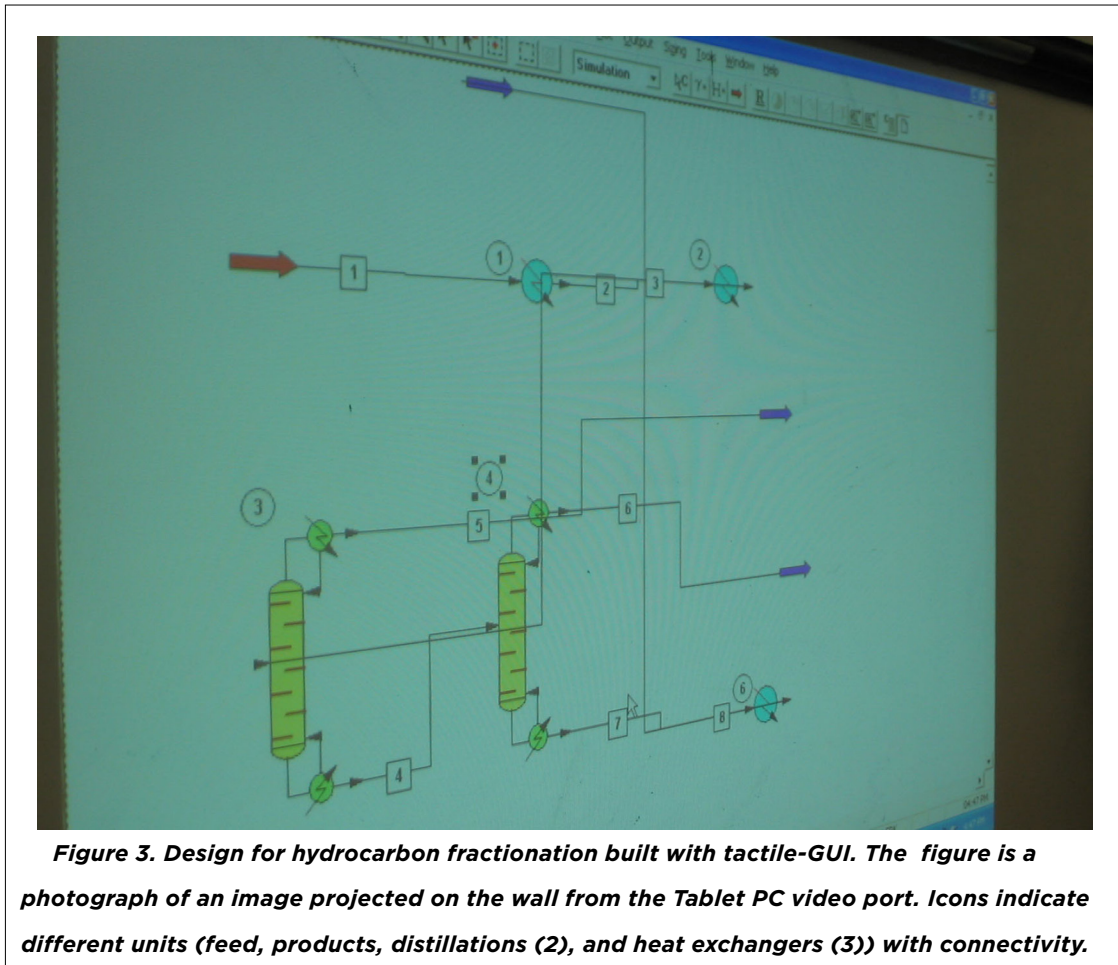

Figure 2. Tablet PC with design medium layer. The computer has a transparency with tactile representations affixed to the material. The Tablet PC has the capability of sensing a left click by tapping and a right click by pushing a button on the barrel of the pen.

Once the visually impaired design student had a tactile layout of the plant on a removable design medium (transparency or screen overlay), he was able to easily find and manipulate tactile representations. Assuming the tactile representations did not move, the student could easily find the [tactile] markers and perform various actions related to design and optimization of the process.

Layout of the plant via the building of the flowsheet involved two major tasks: (i) placing tactile representations of unit operations on the design medium, and (ii) indicating and connecting related unit operations with mouse actions within the Tablet PC (Figure 3). Audible cues embedded in CHEMCAD address the second task. A comparison between users with normal vision and those with limited vision can be made to explain the audible enhancements. A sighted user first decides where to make a connection by placing the cursor close to a port on an icon. The user then starts drawing the stream with a left *click*. Drawing continues until the cursor reaches a different type

Figure 3. Design for hydrocarbon fractionation built with tactile-GUI. The figure is a photograph of an image projected on the wall from the Tablet PC video port. Icons indicate different units (feed, products, distillations (2), and heat exchangers (3)) with connectivity.

of port on another icon, and finally, the user connects the stream with a *left-click*. In contrast, the visually impaired user must first locate the connection port on the tactile representation. The user then listens for the appropriate verbal indicator that signifies either an inlet or outlet port before performing a mouse *click* with the Tablet PC pen.

Initially, reinforcement stickers typically used to strengthen paper around a hole were used to represent unit operations or icons. These stickers were adhesive, making design somewhat intuitive to the user. Since reinforcement stickers had holes in the center, there was a convenient spot that would tactually indicate where the *click* could or should occur, *viz*. a landing or drop point for the screen icon. With the auditory feedback mechanisms built into CHEMCAD, the reinforcement sticker method worked perfectly to connect small unit operation icons such as pumps, feeds, and products, since the drop point for a unit, and the feed and product ports are close together. With the stickers, one only had to search for the sound cue for an inlet or an outlet and drop the pen to *click* and subsequently place the icon on the screen.

The reinforcement sticker method broke down when more complicated unit operation icons such as mixers, distillation columns, or heat exchangers were incorporated into a design. With distillation columns, for example, there were several points where the cursor can be placed to make connections. A sticker with one central point did not achieve fine enough tactile control to allow a distillation column to be attached to a feed point and then discharge the bottoms and the tops of the column into two different parts of the plant. This shortcoming spurred the development of tactile stencils for design of more complex process flow diagrams and computer simulations.

Complex tactile stencils were designed to mimic the same size of the onscreen icons (Figure 4). As a convention, all tactile stencils had a drop point dedicated to the upper left hand corner. Stencils were printed out by using a 1:1 screen shot of the main pallet taken from CHEMCAD, sized to the screen of the Tablet PC, and printed onto adhesive label paper. A careful mapping was made of the click actions associated with the icon drop point and inlet and outlet ports. This was accomplished by overlaying a grid using a built-in CHEMCAD function. Stencils were printed out on adhesive paper and placed on a sturdy material. A heavy acetate polymer that is relatively inflexible had been used as a backing for the tactile stencils. After the stickers were affixed to the backing material, holes were punched where the drop point and ports were indicated. Finally, Braille labels were embossed on the tactile stencil using a Perkins Brailler. The upper right corner of the stencil was then removed, allowing for proper orientation of the stencil on the screen overlay.

Combining the Braille and print on the stencil was advantageous for several reasons. Notably, it enabled a sighted peer to see what the visually impaired student had placed on the design
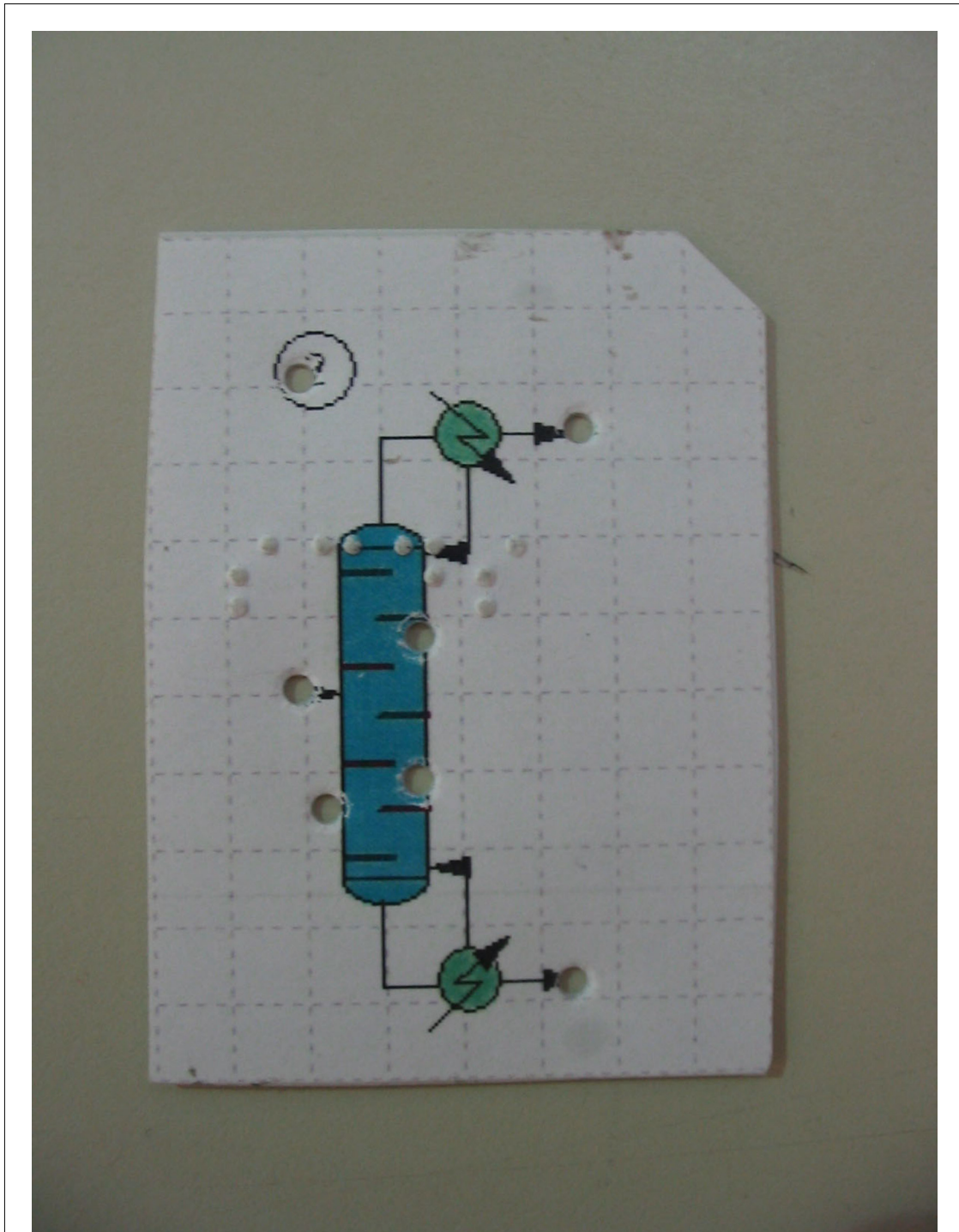
Figure 4. Tactile representation of distillation icon. Note the drop point in the upper left hand corner, and the alignment holes for inlet and outlet ports. Braille labeling is also used to identify icon with type of unit model (SCDS in CHEMCAD).

www.manaraa.com

medium, thereby allowing for an initial troubleshoot of the design before it was transferred to the PC. Alternately, the screen output could be mirrored to an external monitor as was previously noted in Figure 3.
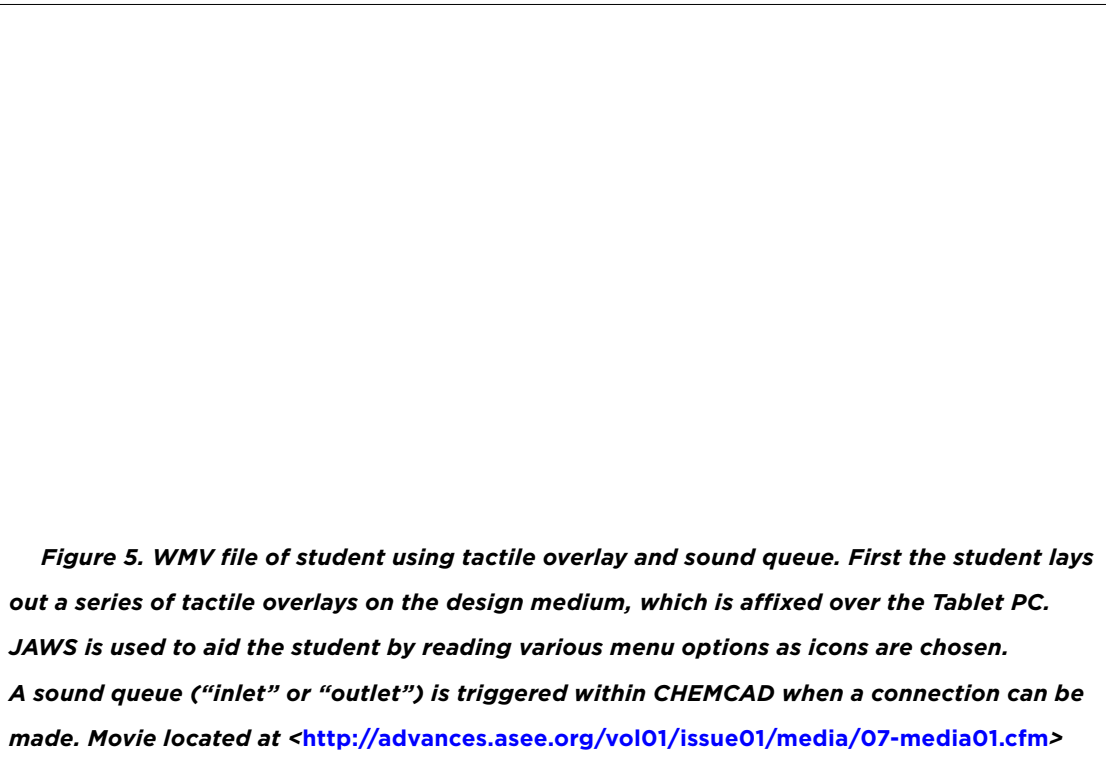
Other problems that had to be solved included the choice of adhesive and design medium that was layered above the Tablet PC screen. Since the visually impaired individual must rely on the fact that the drop points and associated inlets and outlets were in the same place on the screen every time to allow accurate connection of internal streams, it was vital to secure the stencil onto the design medium. Double sided tape was the best choice for adhesion of stencils to transparency. Transparent film was used as the design medium in order to allow a sighted individual to see through to the screen.

Affixed to all transparencies were marked locations of two very important icons not accessible through menu commands: (i) the select unit op submenu icon and (ii) the stream editor icon. A *click* on the tactile representation of the select unit op submenu allowed the user to invoke a submenu which was navigated by the screen reader. This submenu allowed the visually impaired individual to select the unit operation icon of choice without having to search the cluttered visual pallet or place too many tactile elements on the transparency, creating confusion for the student. Another tactile dot marked the stream editor utility that provides connectivity and stitches the process together. These two icons were represented with small, felt covered stickers since a single, simple *click* invoked the respective functions.

Figure 5 is a video file showing complete deployment of the adaptations. The video begins with the visually impaired student placing several icons on the design medium. Once icons were affixed to the medium, mouse-based actions were then used to drop and connect icons whilst employing the audio queues to connect unit operations (feed, pump, heat exchanger, and product). The entire procedure to layout this demonstration took approximately seven minutes. Such a short time is indicative of the utility of this approach since sighted people would likely take about the same amount of time to complete this task.

CONCLUSIONS

The main issue with GUI-based technical software packages is the fact that the student or professional must manipulate graphics with the use of mouse actions to create tangible and working designs. A combination of imbedded audio cues and tactile representations allowed a visually impaired student to operate CHEMCAD, a technical software package used in academia and industry. The adaptations described in this paper could, in principal, be applied to any GUI-based technical package used in a classroom setting. The visually impaired person can easily generate diagrams

Figure 5. WMV file of student using tactile overlay and sound queue. First the student lays out a series of tactile overlays on the design medium, which is affixed over the Tablet PC. JAWS is used to aid the student by reading various menu options as icons are chosen. A sound queue ("inlet" or "outlet") is triggered within CHEMCAD when a connection can be made. Movie located at <http://advances.asee.org/vol01/issue01/media/07-media01.cfm>

with some kind of I/O system such as the Tablet PC as outlined in this paper. Our research shows that using an active design canvas such as the Tablet PC, a visually impaired student can function like his or her visual counterpart in the classroom.

POST SCRIPT

At the time of publication, Mr. Romey is successfully using the adaptations. He is interning at a process design firm, and is tasked with translating current information from written descriptions of Process & Instrumentation Diagrams into working CHEMCAD simulations.

ACKNOWLEDGEMENTS

REFERENCES

[1] Edgar, T. 2000. Process Information: Achieving a Unified View. *Chemical Engineering Progress* 96(1) 51–57.

[2] Pamuntjak, A. The Development of Chemical Engineering Design Computer Programs. <http://www.chemstations.net/documents/devel.pdf>, accessed September 15, 2006.

[3] Turton, R., Bailie, R., Whiting, W., Shaeiwitz, J. *Analysis, Synthesis, and Design of Chemical Processes.* Prentice Hall 1998. Upper Saddle River, New Jersey 07458.

[4] Pictures in a Flash. <http://www.quantech.com.au/products/quantum_products/tactile/piaf.htm>, accessed March 15, 2007.

[5] Raised line drawing kit. <http://www.maxiaids.com>, accessed March 15, 2007.

[6] Suggested Adaptive Tools and Materials for Braille Students in Algebra and Geometry. <http://www.tsbvi.edu/math/tools-blind.htm>, accessed March 15, 2007.

[7] Accessibility at Microsoft. <http://www.microsoft.com/enable/>, accessed October 6, 2006.

[8] Gardner, J., My Vision, Guest Editoral, Journal of Rehabilitation Research and Development, 43 (4), July/August 2006, Pages xiii–xiv, http://www.rehab.research.va.gov/jour/06/43/4/gardner.html, accessed July 1, 2007.

AUTHOR BIOGRAPHIES

Noel Romey is a Graduate Research Assistant in Chemical Engineering at the University of Arkansas. He obtained his B.S. in chemical engineering from the University of Arkansas in 2004. Romey routinely disseminates his work at the CSUN Conference on Disabilities (an international adaptive technology conference). He is a member of Alpha Chi Sigma, Toastmasters International, and the Amateur Radio Club at the University of Arkansas.

Rachel M. Schwartz is a Graduate Research Assistant in the Experimental Psychology Program at the University of Arkansas. She obtained her B.A. (1999) and her M.A. (2007) in Psychology from the University of Arkansas. Her research focus is in the area of cognitive and language development. Schwartz is a member of the International Society for Infant Studies and the Society for Research and Child Development.

Dr. Peter Miao is a Senior Staff Engineer at Chemstations, Inc. He obtained his Ph.D. in Chemical Engineering from the University of Cincinnati in 1996. He has chemical engineering degrees from Texas A&M University (M.S. 1993) and Beijing University of Chemical Technology (B.S. 1982). Miao has an interest in chemical process modeling and engineering software design and development. He has been an AIChE member since 1993.

Dr. Douglas Behrend is a Professor of Psychology at the University of Arkansas. He obtained his Ph.D. from the University of Minnesota in 1986 and his B.A. from Kalamazoo College in 1981. Behrend works in the area of Experimental Psychology, mainly interested in the cognitive and lan-guage development of young children.

www.manaraa.com

Dr. H. Michael Cheung is a Professor of Chemical and Biomolecular Engineering at The University of Akron. He obtained his Ph.D. in chemical engineering from the Case Western Reserve University in 1985. Cheung has M.S. and B.S. degrees in chemical engineering from the Case Western Reserve University (1982) and the Case Institute of Technology (1979). His research interests include engineering education reform, sonochemistry, and supercritical fluids. He is active in ACS, AIChE, and AAUP.

Dr. Robert Beitle is the Louis Owen Professor in Chemical Engineering in the Ralph E. Martin Department of Chemical Engineering. He obtained his Ph.D. in chemical engineering from the University of Pittsburgh in 1993. Beitle holds M.S. (1991) and B.S. (1987) degrees in chemical engineering from the University of Pittsburgh. His interests include biochemical engineering and engineering education. He is an active member in ACS and AIChE.

Address Correspondence to:

Robert Beitle

Louis Owen Professor in Chemical Engineering

Ralph E. Martin Department of Chemical Engineering

203 Engineering Hall

University of Arkansas

Fayetteville, AR 72701 USA

rbeitle@uark.edu