Teaching about Accessibility in Computer Science Education

RICHARD E. LADNER
University of Washington, ladner@cs.washington.edu
STEPHANIE LUDI
University of North Texas, Stephanie.Ludi@unt.edu
Robert J. Domanski
Hunter College (CUNY), robert.domanski@hunter.cuny.edu

Accessibility, in the context of computer science, is about making computing products accessible to people with disabilities. This means designing hardware and software products that can be used effectively by people who have difficulty reading a computer screen, hearing computer prompts, or controlling the keyboard, mouse, or touchscreen. Thus, accessibility topics should be woven into any course about human-facing applications or websites, such as app and web design/development, software engineering, and human-computer interaction. In addition, accessibility is about creating technical solutions to accessibility problems that people with disabilities encounter in everyday living. These technical solutions may include the use of artificial intelligence, computer vision, natural language processing, or other CS topics. Thus, accessibility topics can be included in technical courses, particularly those that incorporate projects where students attempt to solve accessibility problems using techniques taught in the course. There are practical, intellectual, and social reasons to integrate accessibility into computer science curriculum. From a practical standpoint, employers increasingly include accessibility knowledge in job descriptions because they want their products and services to be accessible to more customers and for legal compliance. From an intellectual standpoint, technical solutions to many accessibility problems often require creativity and a multi-disciplinary approach that includes understanding user needs integrated with technical knowledge. From a social standpoint, accessibility is an important topic in addressing inclusivity and an attractive topic for those students who enter the field to do social good, leading to a broader mix of students in terms of gender, race, ethnicity, and ability.

CCS CONCEPTS • Social and professional topics ~ Professional topics ~ Computing Education

Additional Keywords and Phrases: Accessibility, Disability, Teaching Accessibility

ACM Reference Format:
First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 Introduction

The purpose of this article is to help computing faculty members include topics about accessibility and disability in their courses. At the outset we would like to distinguish between "teaching accessibility" and "teaching

accessibly." Teaching accessibly is about the inclusion of students with disabilities in classes. What are the tools and pedagogies that are most beneficial for making sure students with disabilities can learn and succeed? This article is about teaching accessibility, which has to do with the technical content of courses. For example, a web design and development course should have content about the WCAG guidelines and the tools such as ARIA that help make web pages and sites fully accessible to all users, including those who use screen readers or other access tools [18].

Fortunately, there is an organization, Teach Access, that was founded in 2015 with the express purpose of "bringing together industry, education, and disability advocacy organizations to address the digital accessibility skills gap by equipping learners to build toward an inclusive world." All three authors of this article are active members of Teach Access. One important activity of Teach Access is their Faculty Grants program that provides financial support to faculty members who want to incorporate accessibility topics into their courses. We encourage readers of this article to visit the Teach Access website to learn more about the organization. At the end of this article we will point out many resources where computing faculty can learn more about what they can include in their courses about accessibility.

1.1 What does Accessibility Mean?

In the context of computing, accessibility means having the ability to access computer applications, websites, and other human facing artifacts by people who have differing abilities. Some users might be blind or have low vision, deaf or hard of hearing, have limited mobility, or have other conditions that may limit their ability to access computers using conventional access techniques such as a visual display, keyboard, mouse, or touchscreen gestures. Instead these users may use alternative input and output, such as screen readers, magnification, speech input, switch input, sip and puff, eye gaze, and more. Most of these alternative input and output methods are supported in the standard operating systems provided by Microsoft, Apple, and Google. However, unless a computer application or website is properly programmed to take advantage of the accessibility support, they will not be accessible. In addition, unless accessibility is incorporated into applications and websites early in development, it is often difficult to impossible to add accessibility in as an afterthought. Nonetheless, there are some easy parts of making accessible applications and websites. For example, making sure that text and its background have high contrast is beneficial for people with certain eye conditions, avoiding blinking elements on a screen is beneficial for people who have certain triggers for epilepsy, and adding alternative text to images is beneficial to blind people.

Beyond making applications and websites accessible, accessibility includes the development of applications and other tools that make life easier for people with certain kinds of disabilities. For example, a screen reader is an application in its own right. It converts text to speech and supports navigation within an accessible application or website. There is an entire industry segment, often called the assistive technology industry sector that focuses on technology to improve the lives of people with disabilities. This industry produces thousands of products from power wheelchairs to sip and puff devices. This industry sector includes large companies such as Microsoft, Google, Meta, and Apple that have pioneered some of these technologies. Examples include the XBox adaptive controller from Microsoft, automatic captioning for YouTube from Google, and touchscreen screen reader VoiceOver from Apple.

Accessibility also has an active research community that includes industry and academia worldwide. Accessibility research appears in the ACM SIGACCESS sponsored annual ASSETS conference which is all about accessibility. Other ACM sponsored HCI conferences, CHI, UIST, and CSCW often have sessions on accessibility related papers. Other conferences that feature accessibility research and development are: RESNA, ICCHP, ICCAAT, and CSUN. There are also a number of conferences that are primarily about assistive technology development including Closing the Gap, Accessing a Higher Ground, and the Assistive Technology Industry Association Conference. Within academia there are hundreds of individual researchers

who are active in accessibility research and development. There are a few centers of excellence including the Center for Research and Education on Accessible Technology and Experiences (CREATE) at the University of Washington, the Trace Research and Development Center at the University of Maryland, and the Coleman Institute for Cognitive Disabilities at the University of Colorado.

1.2 Need for Teaching Accessibility

There are at least four reasons for including accessibility in the computer science curriculum: industry demand, an intellectually challenging computing topic, equity, and broadening participation. The demand by industry is well documented by the Partnership on Employment & Accessible Technology (PEAT) that fosters collaborations to help make workplaces more accessible. They report in a study that 57% of businesses cannot find employees with the accessibility skills they need [11]. The companies that support the Teach Access organization agree that they would like to be able to hire more employees who are already versed in accessibility.

Accessibility is an intellectually challenging topic from learning about the needs and technologies of end users with disabilities to the intricacies of operating systems functioning in the presence of access technologies that may override standard input and output functions. There is also the challenge of making entirely new technologies such as those found in augmented and virtual reality devices accessible. The accessibility research community is continually finding new ways to use technology to improve the lives of people with disabilities.

According to the World Health Organization there are about 1.3 billion people in the world who have a disability. There are hundreds of millions of people in the world who are blind or low-vision, are deaf or hard of hearing, have a mobility related disability, or have a hidden disability such as dyslexia or other mental or emotional condition. Disability could be congenital, be the result of disease or accident, a byproduct of aging, or be the result of war. In terms of equity this group deserves access to computers and their applications and websites. The UN Convention on the Rights of Persons with Disabilities (CRPD) addresses what countries should do to assure the rights of people with disabilities. Article 9, out of 50, addresses what the technology sector should be doing.

> *Promote access for persons with disabilities to new information and communications technologies and systems, including the Internet. Promote the design, development, production and distribution of accessible information and communications technologies and systems at an early stage, so that these technologies and systems become accessible at minimum cost.*

Additionally, ABET's proposed changes for the 2024-2025 Criteria for the Accreditation of computer science programs has a new emphasis on accessibility. It uses the term DEIA for diversity, equity, inclusion, and accessibility to describe the importance of including the widest groups of students and faculty in computing departments. It also proposes to add accessibility topics that must be included in the curriculum.

> *Principles and practices for developing inclusive and accessible computing solutions that address the needs of diverse user communities.*

Thus the CRPD and ABET both call for teaching accessibility because it directly promotes equity.

Computing fields as a whole are concerned about broadening participation, meaning bringing in more diverse populations into the fields. This means bringing in more women, Black, Indigenous, and People of Color (BIPOC) people, disabled people, and members of other minoritized groups into the fields. One way to do this is to have a computing curriculum that supports "computing for good" such as accessibility. Disability is one of

those attributes that cuts across all other demographics, meaning that people in all these minoritized groups can identify with disability. Accessibility is a computing for good topic that can attract individuals from minoritized groups to computing fields [17].

1.3 Paper Outline

In the remainder of this article we delineate the major disability groups and the kinds of technology they use for accessibility (Section 2); the major accessibility guidelines (Section 3); the importance of accessibility research (Section 4); and examples of what to teach about accessibility in several popular CS courses including Intro to CS (CS0), web development and design, human-computer interaction, software engineering, AI and machine learning, and project-based learning in capstone courses (Section 5).

2 Disability and Accessibility Detailed

Before teaching any course that includes accessibility as a topic it is important to talk about people with disabilities and the affordances they use to access computers. This will help students appreciate the accessibility topic that you are presenting in your course.

2.1 Types of Disabilities and Their Impacts

The disability spectrum is diverse and comprised of groups of people that are as diverse as other marginalized populations. In this section a brief overview of the broad types of disabilities are presented to provide context for the rest of the paper. It is important to note that within each group described in the following sections there is a wide spectrum of abilities and challenges. Additionally, disability can cut across types of disability. The types of disability described in this section will be discussed in terms of their impact on CS Education rather than from a medical perspective. Discussion of the assistive technology used can be found in Section 2. 2.

2.1.1 Blindness and Visually Impaired

Computer Science students who are blind or have low vision can have various challenges depending on the degree of visual impairment and their access needs. Access to course materials is critical, to where access to materials that are themselves compatible with screen readers or other assistive technology is critical. Programming tools and other course-related tools also need to be accessible to the student or alternatives located. In class, announcements should be provided in different modalities to ensure that all students understand any changes or reminders. Many computer science courses include lectures with material written on the whiteboard. As such, the material needs to be captured and conveyed to students in a manner that enables participation. The best way is to provide the material in advance though some students have assistive technology or personnel to provide access in real time. When lecturing, it is best practice to be clear in your speech to convey the sequencing and specificity of the material at hand (e.g. avoid just circling things on the board or pointing to something without elaboration).

2.1.2 Deaf and Hard-of-Hearing

Deaf/Hard-of-hearing. Persons who are Deaf/Hard-of-Hearing need to be able to effectively communicate with their instructors, classmates, and supporting staff regarding the course material and expectations. Some students may be able to read lips, which requires instructors to consistently face the student as they speak. Some students may have a sign language interpreter or assistive technology that captures verbal information and presents it in a manner that the student can access it. When an interpreter is in the class, make sure that

there is room for them to sit in front, near the student. When instructors communicate with students who are Deaf/Hard-of-Hearing, the instructor needs to look at the student, not the interpreter. To support the interpreter and the student, slides or other materials should be provided in advance of the course. When slides have content, the interpreter is trying to convey both the technical content and what the instructor is saying to the student at the same time. That being said, the instructor should be careful to speak clearly and at a reasonable pace. Students whose primary language is sign language (ASL or otherwise) may have issues with communicating in writing (e.g. grammar, word choice) as the primary language of the locale may be a secondary language to the student. Knowing that, you should consider how you grade the quality of writing on exams or other course assignments for students whose first language is not that of the locale of the course. Lastly, when videos are used as part of a course, there should be closed-captions for the video so that the content is accessible.

2.1.3 Learning Disabilities

Persons with learning disabilities can benefit from a variety of accommodations, which can in turn universally support students broadly. Like other types of disabilities, there is a wide spectrum of abilities and challenges represented by learning disabilities. For example, some students have difficulty processing auditory-based information while others may find visual information challenging. In other cases, students have difficulty sequencing steps in an activity or to plan for them. Accommodations that can support students with learning disabilities can support all students. Examples include: providing slides or other course materials ahead of time, clearly formatting instructions so that they are easy to follow and keep track of, provide announcements and course material in multiple modalities, to provide clear expectations on how and when to ask for help and submit work, as well as how work will be graded (e.g. rubric or grading criteria). By providing these types of infrastructure, students can more readily follow the course and your expectations.

2.1.4 . Mobility-Related Disabilities

Persons with mobility challenges can face issues with accessing educational spaces from the building itself to fine grained classroom layout issues. Additionally, desk height can be a mismatch to the desk height required by their wheelchair, scooter or other mobility aid. Dexterity challenges can be represented by issues with computer input devices in labs or similar handheld devices or material handling that is needed as part of their courses. In an instructional setting where the computer is used, the student's computer should be allowed to be used though support by classroom computers are important as well. For instances, where team work is needed (e.g. presentations, exercises) all students should have the access needed to be successful.

2.1.5 Neurodiversity

Students who are neurodiverse are the most likely group of students with disabilities (in addition to students who have learning disabilities) to participate in Computer Science programs. Research has shown that students with Autism are attracted to Computer Science and related disciplines [3]. Accommodations needed to support neurodiverse students will also support students who are neurotypical. Examples include: providing an overview for the class meeting so that the order of topics and activities is known, providing a course overview in the syllabus with clearly delineated due dates and instructions on assignments submissions, etc., and being clear with class expectations in areas like asking questions, behavior, and exams. Some students, especially when in a large or noisy environment, will need to be able to leave class to address an overwhelming or over stimulating environment. Many students also benefit from access to class materials, including slides, in advance. For courses where a teamwork component is required, a class discussion on how to best communicate during meetings or online can maximize student success.

2.1.6 Temporary or Situational Disabilities

When most people consider disability, permanency is assumed. From a design or instructional perspective, disabilities that are temporary or as a result of the environment or situations also necessitate accommodation. Examples of a temporary disability are a broken arm after an accident or difficulty in standing extended periods when pregnant. Situational disabilities can be contextual such as the inability to hear certain sounds in a noisy environment (e.g. a nightclub or factory). These and other temporary and situational disabilities can be identified and accounted for during system development.

The above categories of disability are not a complete list. Instead they highlight those groups that may have access issues in technology and services. Examples of other types of disability include health-related disability and mental health diagnosis, which can be associated with invisible disabilities. It should also be noted that disability is also associated with aging. While not every person has a disability as they age, many people do face a variety of challenges and limitations as they age including those which are considered disabilities.

2.2 Typical Affordances Used by People with Disabilities to Achieve Access

In addition to the context of what types of disabilities exist, it is important to note different approaches used by persons with disabilities as they access information and services. The list below is not exhaustive though it provides foundational knowledge in the domain.

2.2.1 Screen Readers

Screen readers provide voice-output where the software reads the contents of the screen to the user, as well as supporting interaction through content focus and supporting navigation and interaction through keyboard shortcuts [1]. The depth of information depends on the design of the software, app, or website so that the users can understand content, its structure, and navigate the user interface. Many persons who are blind and visually impaired use screen readers, as well as some people with Dyslexia or some learning disabilities. Examples of screen readers include Apple's VoiceOver for MacOS and iOS, NVDA (open source), and Freedom Scientific JAWS.

2.2.2 Keyboard Input

While gestures and mouse-based input are common, some people rely on the keyboard as their primary input device when using technology. The keyboard can be a hardware device or an on-screen keyboard. For example, blind and visually impaired users who use screen readers typically use the keyboard as their input device. Other persons who primarily use the keyboard as their affordance are persons with dexterity issues where using a mouse is a challenge and those who may use a stylus to select keys on the keyboard for a variety of reasons. The use of the keyboard with autocompletion features can support persons with limitations in communication.

2.2.3 Switch Control

A switch is a way for a person to activate a control as part of a task. There are many types of switch controls, from a large, oversized button to facilitate the activation of a control by a person’s foot or elbow to a small switch that can be activated in specialized ways such as by a person’s cheek or head movements. Coupled with the ability to move the cursor on-screen, users with limited hand or body movement can access software and

information. User interfaces that can be controlled by the keyboard are amenable to working with switch controls.

2.2.4. Speech Control

Voice-based input or speech-control are used by varied groups of people. Persons who are quadriplegic or who have dexterity issues related to Cerebral Palsy or other conditions, use speech-based input to interact with software and services. Some systems require training so that they can be calibrated to a particular user while others contain a specific set of commands or keywords that are used for interaction.

2.2.4 Input Alternatives

In addition to the input devices described in prior sections, other alternative input exists to address the needs of various users who are not able to use a traditional keyboard and mouse. Examples include one-handed keyboards, keyboards with large keys or alternate configurations, joysticks, trackballs with oversized balls, and eye/motion tracking devices.

The assistive technology discussed in these sections are common categories of assistive technologies in use. As with other areas of Accessibility research, new interfaces are under development or being refined in order to be more effective and efficient. For example, advances in Machine Learning and Natural Language Processing are being explored as means to increase access to language for those who have challenges with communication. As technology continues to evolve so does assistive technology.

3Accessibility Standards

Accessibility standards have been in place since at least 1973 with passage of Section 508 of the Rehabilitation Act. In 1995, the World Wide Web Consortium's Web Accessibility Initiative (W3C, WAI) came out with the first Web Content Accessibility Guidelines (WCAG). They have been updated over time to cover web-based interactive applications. Inherently, they also cover the accessibility of native applications, not just web pages. The European Union also has its own accessibility standards that generally have more legal force than the more voluntary standards.

3.1 WCAG 2.1

The W3C's Web Content Accessibility Guidelines (WCAG) serve as the canonical set of recommendations for making web content more accessible – with conformance addressing the needs of people with a wide range of disabilities including blindness and low vision, deafness and hearing loss, limited movement, speech disabilities, and photosensitivity. Additional criteria to address cognitive, language, and learning disabilities are also under development.

The WCAG guidelines are based on four principles that provide the foundation for web accessibility: that content is perceivable, operable, understandable, and robust. Serving as a shared standard designed to meet the needs of individuals, organizations, and governments internationally, testable success criteria are provided for each guideline allowing for WCAG to be used wherever requirements and conformance testing are necessary - such as in design specification, purchasing, regulatory compliance, and contractual agreements.

For practitioners, see the Quick Reference on "How to Meet WCAG" requirements and techniques.

3.2 EU: EN 301 549

The European Union's "Accessibility requirements suitable for public procurement of Information and Computing Technology (ICT) products and services in Europe" is the harmonized standard developed in response to a request by the European Commission. It specifies the functional accessibility requirements applicable to various types of ICT products and services for organizations engaged in public procurement within the European Union, and also provides a description of the test procedures and evaluation methodology for each requirement.

The standard addresses a wide range of ICT products and services that goes far beyond the scope of only addressing the accessibility of web content. It covers web-based technologies, non-web technologies, and hybrids that use both, and also sets forth requirements for various categories of hardware, software, and services including both audio- and video- based ICTs, biometrics, documentation and support services, and more.

3.3 US: Section 508 standards and Section 255 guidelines

Similar to the EU's 301 549 standard, in the United States the federal standard for ICT accessibility relating to procurement is set forth in Section 508 of the Rehabilitation Act and the guidelines set forth in Section 255 of the Communications Act. These require that all ICT "developed, procured, maintained, or used by federal agencies" ensure access for people with physical, sensory, or cognitive disabilities. It covers various forms of hardware, software, telecommunications equipment, multifunction office equipment, and web and digital assets to be accessibility-compliant.

The U.S. Access Board – an independent federal agency charged with promulgating these standards – provides technical assistance through webinars and trainings, an ICT Testing Baseline for web accessibility, and Functional Performance Criteria mapped back to WCAG Success Criteria.

4 Accessibility Research

It is important to recognize that accessibility is not a static subject, but is constantly changing because of advances in technology and the ingenuity of researchers to come up with new ideas to employ existing technologies to make the lives of people with disabilities better. This dynamic nature of accessibility is no different from the dynamic nature of most subfields of computer science where we as teachers have to continuously update our courses and curricula when they get out of date faster than we would like.

A good example of this dynamicity came out in the early 2000's when touchscreens became popular with the invention of the iPhone. The blind community was upset with this new product because it was inaccessible; it had only one button that didn't do much except return the phone to the home screen. Members of this community were very familiar with feature phones that had buttons for doing various actions. Some feature phones were marketed directly to blind users because of advanced features that made them more accessible. A research project led by Jacob Wobbrock and his students Shaun Kane and Jeff Bigham took notice and came up with a novel idea that would make the iPhone accessible [9]. They developed Slide Rule, a touchscreen app that could implement a home screen with phone, email, and music player apps. Accessibility was accomplished by having the app talk out loud what is touched with a one-finger scan of the touchscreen. Various touchscreen gestures were supported including flicks up, down, left, and right to navigate an array of buttons on the screen and a split-finger tap to activate a button that is touched. This was the first demonstration that a

touchscreen device could be accessible. About a year later Apple introduced the iOS VoiceOver screen reader that made the iPhone accessible. Some of the gestures from Slide Rule ended up being part of VoiceOver.

Some research that is not labeled as accessibility research has had a huge impact on accessibility over the years. Examples of this include optical character recognition (OCR), speech synthesis, spoken language recognition, and video compression. Early OCR and speech synthesis research led to the Kurzweil Reading Machine that gave blind people the ability to read printed books and articles [15]. Spoken language recognition which has improved greatly in the last ten years gives deaf people access to speech via automatic captioning. Modern video compression technology allows deaf people to communicate in sign language remotely on their smartphones. It is important to highlight in computer vision, natural language processing, and signal processing courses the accessibility solutions that arise from these disciplines.

5Ways of Integrating Disability and Accessibility into the Curriculum with Brief Examples

In the following subsections we give some advice on how to potentially integrate disability and accessibility into several courses. A more detailed and comprehensive way to do this is found in the new online book [12]. In preparing for including accessibility into courses it is important to think about how the course can be taught accessibly. So some advice is given on teaching accessibly so that the course itself models what it is teaching.

5.1 Intro Sequence, Including CS0 or Similar Courses

The introductory Computer Science courses provide students with foundational problem-solving and programming skills that they will then build upon as they traverse their program of study. This sequence of courses often focuses on a specific programming language and covers the basics of programming syntax, constructs, and structures. At this stage, students are often learning new tools and processes, not just the programming skills. It is important to have concrete examples that students can relate to as motivation for learning. Accessibility is one of those examples. These courses have several assignments, labs, and/or recitations to offer students an opportunity to practice what they have learned and to apply it to a problem.

Some areas to consider and topics to include in the introductory course are listed below. The low-level details of the technical topics themselves may vary depending on the languages taught, however some suggestions are provided.

Course Planning

- Conveying the structure of activities and scaffolded content within a class meeting and through the progression of the course will support learning for all students. Neurodiverse students, as well as those who may have challenges with executive functioning or anxiety, will appreciate the structure and expectation of what is coming next. In these introductory courses, students who are new to the discipline can also benefit from the clear communication and structure.
- Select tools that are accessible so that students with disabilities can succeed in the course(s). Console-based tools are still common in many introductory courses; whereby accessibility is often easier when compared to more complex, GUI-based development environments.
- Ensure that assistive technology (e.g. screen readers) are installed on lab computers for students who need access.

- If online tutorials are used, select those that are accessible. Any textbooks used, should be identified in advance so that students who need to get them in an alternative format have time to do so.
- Train Teaching Assistants in working with students with disabilities. Your campus office that supports students with disabilities can be a resource for your department.

Topics Where Accessibility Can Be Integrated [5]

- String-based information should be used to emphasize their relevance in presenting information to persons who use screen readers.
- Introducing the habit of considering accessibility in program design and implementation early (as a good habit) as opposed to an add-on at the end. This is similar to good practices for security.
- Sound output can be added to the visual output of a program, especially as a basic tool to indicate the traversal of code or a loop.
- Integrate the use of user interface layout managers or similar tools into assignments/projects to help students understand the need to properly group UI elements and their impact on the traversal of interface for screen readers and tab/keyboard navigation.

5.2 Web Development and Design

Web development and design courses are rich places to introduce accessibility because web accessibility has been a focus of the WCAG guidelines since they were first introduced in 1995 not long after the World Wide Web and modern web browsers became popular. In addition, there are modern tools such as the Accessible Rich Internet Applications (ARIA) suite of web standards that supports the accessibility of interactive web pages that depend not only on HTML and CSS, but Javascript, and other programming tools to create interactive web pages. One thing to do at the start of the course is to announce course goals which should include a discussion of best practices in developing web sites including those related to accessibility. It would be helpful for students to understand how people use the web with alternative technologies such as screen readers, speech input, and switches. Students should understand that the web is for everyone regardless of their disability. Most existing web design and development courses can be modified to include accessibility topics. Here are some ideas for course planning and accessibility topic integration.

Course Planning

- Demonstrate the value to students of incorporating accessibility design principles into websites that they build; Provide evidence to students of industry demand relating to hiring candidates with such skills.
- Learn about the fundamentals from the W3C-WIA curriculum framework.
- Choose web development tools that support accessibility integration.
- Review the syntax for how to incorporate common accessibility-enabling code within HTML
- Identify quality automated web accessibility checkers.
- Develop short demos of how screen readers, speech input, and switches work on the web.

- Develop a “Think-Pair-Share” in-class activity for students to explore various personas in their design thinking process, similar to the “GenderMag” method for evaluating software inclusiveness [2].
- If possible, invite guest speakers from industry who have experience with accessible web design and development. They may be willing to give a lecture about how they incorporate accessibility in their own workflow.

Topics Where Accessibility Can Be Included

- Discuss how to make sure there is adequate contrast between text and background.
- For various elements on a web page, images, lists, tables, buttons, and other widget discuss how they can be made accessible then expect that the accessible versions and design practices are applied in course work.
- Discuss how to use ARIA to make interactive pages accessible.
- Give web design and development projects that require accessibility testing. Students can turn in the results from automated accessibility checking along with the developed web pages.
- Demonstrate how search engines rank accessible websites higher in their results and explain the value of this in terms of SEO strategies

5.3 Human-Computer Interaction

Due to the nature of Human-Computer Interaction courses, there are opportunities to integrate accessibility into the courses and provide students with foundational knowledge and design skills that they build upon in other courses or Capstone projects. The HCI course sometimes focuses on prototypes while others include a finished project. Some offerings focus on Web or Mobile development while others are more general in nature in terms of technology paradigms or platforms. At this stage, students are often learning new tools and processes in areas such as User Experience (UX) design and usability testing. These courses may have several assignments and/or individual or team projects that offer students an opportunity to practice what they have learned and to apply it to a problem.

Some areas to consider and topics to include in HCI or related courses are listed below. The low-level details of the technical topics themselves may vary depending on the technologies or tools taught, however some suggestions are provided.

Course Planning

- The suggestions in Section 5.1 come to bear here, however non-computer-based tools can also be leveraged. For example, prototyping with tactile objects may be useful, as well as rapid prototyping tools that support annotation can be useful.
- Include persons with disabilities in the demonstration of assistive technology use or to provide feedback during the course project can bring real world perspectives into the course.
- Invite guest speakers to discuss prototyping strategies and tools, as well as to reinforce the relevance of accessibility to the “real world.”
- Include expectations for accessibility into course projects, as well as modifying project templates and grading rubrics to reflect the expectations.

Topics Where Accessibility Can Be Integrated

- Usability testing can include persons with disabilities at different stages of development (e.g. cognitive walkthrough, formal usability testing)
- Introduce the inclusion of considering accessibility in the development process used (e.g. user-centered design) as opposed to an add-on at the end. This is similar to good practices for other cross-cutting system qualities.
- Teach the use of Web Accessibility testing tools for web-based projects, as well as relevant platform standards for accessibility.
- Integrate the use of user interface layout managers or similar tools into assignments/projects to help students understand the need to properly group UI elements and their impact on the traversal of interface for screen readers and tab/keyboard navigation.
- Teach the development and use of personas that include persons with disabilities.

5.4 Software Engineering

The focus of undergraduate Software Engineering courses is to provide a broad overview of the process and activities needed to deliver software. Some undergraduate programs have a single course while others may have a course sequence that provides additional material and project work. The general expectation is that students are fluent in at least one programming language and have completed some other required courses to help provide some maturity in their Computer Science skills. Software Engineering courses are often upper-division courses, though some programs offer Software Engineering as early as the second year.

Project work in a small team is commonplace in Software Engineering courses, though some programs focus on the theory rather than the application of Software Engineering. For courses that include a team project, the opportunity exists to include the expectation of accessibility within the project. Due to the timing of projects, especially when there is only one course in Software Engineering, the scope of the project is often defined by the instructor. The instructor-led projects often target specific technologies, a familiar domain, and scope that fits into the course schedule. These projects can often be easily modified to include accessibility requirements, even if for a specific population of persons with disabilities. Due to the introductory nature of the courses and many students having had limited team-based and/or long-term project-based experience, having specific requirements can help keep the teams focused. Since the instructor will be well-versed with their project, they can identify someone to provide some initial expertise in the accessibility space for the domain. For example, a colleague or someone from the community can be a guest speaker and discuss their challenges in the domain space or they can offer some time to answer student questions at different points during the term (including testing). The use of accessibility standards and testing tools is also very useful at this stage as the students can do some level of initial retaining on their own. At the onset of the project, students should conduct research as to existing approaches to accessibility as it relates to their project and domain so that missteps or duplicated work are avoided. For example, many students are under the impression that they must provide voice output for their apps to enable use by persons who are blind when the project needs to work with the platform's screen reader (using appropriate design practices).

Due to the often-advanced nature of the course, the course is also an excellent opportunity to reinforce material learned in prior courses. The topic can be revisited not only as each major software engineering phase is presented, but also as the lifecycle is traversed (e.g. waterfall, iterative, agile).

For projects with externally sponsored projects, you should refer to the section that discusses Capstone projects. Software Engineering courses which focus on the theory without the project can instead integrate accessibility into the assignments since that will provide the individual practice for the material.

Some areas to consider and topics to include in Software Engineering or related courses are listed below. The low-level details of the technical topics themselves may vary depending on the technologies or tools taught, however some suggestions are provided.

Course Planning

- The suggestions in Section 5.1 come to bear here, however non-computer-based tools can also be leveraged. For example, prototyping with tactile objects may be useful, as well as rapid prototyping tools that support annotation can be useful.
- Include persons with disabilities as part of requirements elicitation and to demonstrate assistive technology use or to provide feedback during the course project can bring real world perspectives into the course.
- Include expectations for accessibility into course projects, as well as modifying project templates and grading rubrics to reflect the expectations.
- Include links to Accessibility Guidelines and Specifications for the platforms and technologies used in the course.
- Convey the expectation that project demonstrations also show the degree of accessibility.
- Include links to terminology used in the disabled community so that project artifacts and other project documents and the project itself conveys inclusion with current terminology.

Topics Where Accessibility Can Be Integrated

- Include the expectation Test Case Suites include test cases for accessibility.
- Follow usability testing practices that include persons with disabilities at different stages of development (e.g. cognitive walkthrough, formal usability testing)
- Introduce the inclusion of considering accessibility in the development process used (e.g. user-centered design) as opposed to an add-on at the end. This is similar to good practices for other cross-cutting system qualities.
- Teach the use of automated Web Accessibility testing tools for web-based projects, as well as relevant platform standards for accessibility.
- Integrate the use of user interface layout managers or similar tools into assignments/projects to help students understand the need to properly group UI elements and their impact on the traversal of interface for screen readers and tab/keyboard navigation.
- Teach accessibility as part of the software engineering project, not an add-on at the end. As such discussion occurs from requirements through deployment and maintenance.

5.5 AI and Machine Learning

Modern artificial Intelligence (AI) and machine learning (ML) courses often cover their algorithmic theories, but also some of their applications and social implications. Of particular interest to many students are applications for social good some of which could address accessibility. Additionally, computer vision (CV) and natural language processing (NLP) also cover AI and ML, but with more focus on their applications.

Course Planning

- The suggestions in Section 5.1 come to bear here. In these courses mathematical approaches are important. Mathematical notation can be quite challenging for people with some disabilities such as

blindness. If you know of such a student who will be in the class, make sure they have the tools for math accessibility.

- Do some preparation to find machine learning applications that help solve some accessibility problems to use them as examples.
- Major social concerns about AI and ML applications have to do with inclusivity and bias [11]. Inclusivity means that it works well for a variety of populations. The opposite is bias which typically means that the data used to train the application is not large or varied enough so it works well for some people. For example, an ML application that recognizes pedestrians in a self-driving car may not be able to recognize someone in a wheelchair unless wheelchair users are included in the training set.

Topic Where Accessibility Can Be Integrated

- An AI or ML course could address the problem of automatically finding a good accessibility setting for a computer user who has a disability based on a series of tests or answers to questions. Some early work on this was in the development of Supple, an AI-based application that helped create an accessible user interface for someone with low vision or mobility related disability [7]. More generally, the concept of ability-based design presents the challenge of building a user interface that is accessible automatically from analyzing the abilities of the particular users [19].
- A CV course could cover some of the applications of CV to accessibility such as finding text in images and optical character recognition. These are the basis of accessibility applications to assist blind people in their environment. A NLP course can cover spoken language recognition which is the basis of automatic captioning of videos and applications like Android's Live Transcribe for speech recognition. These are useful for deaf and hard of hearing people.
- NLP has great potential for improving the lives of people with disabilities. Speech recognition and synthesis are already found in many access technologies, such as screen readers and voice control systems, both of which are used heavily by disabled people.

5.6 Project-Based Learning within Capstones

Many undergraduate programs culminate in a Senior Capstone course or course sequence that provides students with the opportunity to work as a team on a long-term project. The pedagogical placement of such a project allows for students to both bring together the knowledge and skills learned during their undergraduate program and to learn new skills and knowledge as required for the specifics of the project itself. Some projects are devised by the capstone instructor while others are gathered from outside project sponsors (e.g. non-profits, start-up companies, established companies).

The capstone course or sequence for undergraduate Computer Science students is commonplace in many undergraduate programs [6]. For ABET accredited programs, Capstone experiences satisfy the requirement that students "solve open-ended real-world problems in their respective engineering disciplines prior to graduation" [14]. Team projects are also a central aspect of many capstone courses, which may be a single term or last 2 terms in duration. Often the course instructor oversees the course in terms of providing needed instruction as well as managing all the capstone projects (including recruiting sponsors and vetting projects).

The capstone projects can be general projects or specialized for users with disabilities. General projects that are accessible can be preferred since their accessibility to persons with disabilities promotes access to said services or information covered in those systems. Best practices from research of engineering teams working on capstone projects for persons with disabilities [8] informs this section. Examples include using a user-focused process rather than solution-oriented approach and transparency between the client, team, and advisor. Recent work by researchers studied instructional interventions and their short and long-term impacts [8, 10, 14], providing further foundational knowledge in design for users with disabilities in the context of Universal Design as many systems are used both by users with and without disabilities.

When considering accessibility, as in other quality attributes, attention is needed throughout the development lifecycle. For capstones that build upon a Software Engineering course, connections can be made to material covered in the course that is then expanded upon in Capstone. Regardless of projects pitched as being for persons with disabilities or for a general population, there are opportunities for integrating accessibility throughout the process. Best practices include:

- Setting the expectation that accessibility is part of the system, even when the system is not explicitly for persons with disabilities. Examples include cooking apps, study aids, and social apps.
- During the requirements elicitation activities, persons with disabilities should be included along with their non-disabled counterparts. If surveys are distributed, the use of an accessible platform and distribution on forums or social spaces for persons with disabilities is important.
- Persons with disabilities should give feedback on the design and later on working versions of the system as part of the process. Depending on the project itself, a small group can give feedback to a small number of capstone projects via a type of “office hours” or through some other mechanism. Many systems may be able to gather feedback remotely.
- Design of software systems using accessibility standards, which students can then conduct their own testing against with appropriate tools.
- Evaluate technologies for their support of accessibility, as well as other needs.
- Testing should include the use of accessibility testing tools and assistive technology, where relevant. However, these tools are not a substitute for testing by and feedback from persons with disabilities.

The support of such approaches when paired with inclusive teams, can result in a system that is intentional with accessibility. Systems that are specifically needed by persons with disabilities are also worthy projects. Such projects effectively force students to design a system for persons that they are most likely unfamiliar with. While these projects are impactful and can provide an effective instructional experience, the need to teach the importance of making general software accessible is an important lesson in itself.

6Conclusion

Accessibility is an opportunity for computing programs. An opportunity for educators and students to enhance their skill set and apply these skills in various courses that will serve graduates as they embark on their careers. The discussion of the integration of accessibility into a variety of courses serves as the starting point for faculty to reflect on course content, as well as those of curriculum committees when viewing program curricula from a higher level. The standards referenced in this paper, as well as breakthroughs in research, continue to evolve as needs change and technology advances. These areas serve as opportunities for departments also. In the end, these initiatives inform innovations and improvements that not only serve persons with disabilities but also the varied needs of all people in the ever changing contexts of accessing information and services.

ACKNOWLEDGMENTS

The preparation of this article was supported by NSF Grant No. 0540615. Thanks to Caitlin Malone and Julian Brinkley who gave us valuable feedback on an early version of the paper.

REFERENCES

[1] American Federation of the Blind. Screen Reader. Retrieved October 2, 2022 from: https://www.afb.org/blindness-and-low-vision/using-technology/assistive-technology-products/screen-readers

[2] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, William Jernigan. November 2016. GenderMag: A Method for Evaluating Software's Gender Inclusiveness. Interacting with Computers, Volume 28, Issue 6, 760–787. https://doi.org/10.1093/iwc/iwv046

[3] Catherine Caldwell-Harris, & Chloe Jordan. 2014. Systemizing and special interests: Characterizing the continuum from neurotypical to autism spectrum disorder. Learning and Individual Differences. Volume 29, Issue 2014, 98-105. https://doi.org/10.1016/j.lindif.2013.10.005.

[4] CAIR: RIT Center for Accessibility and Inclusion Research. Retrieved September 7, 2022 from http://cair.rit.edu/projects.html

[5] Robert F. Cohen, Alexander V. Fairley, David Gerry, and Gustavo R. Lima. 2005. Accessibility in introductory computer science. In Proceedings of the 36th SIGCSE technical symposium on Computer science education (SIGCSE '05). Association for Computing Machinery, New York, NY, USA, 17–21. https://doi.org/10.1145/1047344.1047367

[6] Robert F. Dugan Jr (2011) A survey of computer science capstone course literature, Computer Science Education, 21:3, 201-267, https://www.tandfonline.com/doi/abs/10.1080/08993408.2011.606118

[7] Krzysztof Gajos and Daniel S. Weld. 2004. SUPPLE: automatically generating user interfaces. In Proceedings of the 9th international conference on Intelligent user interfaces (IUI '04). Association for Computing Machinery, New York, NY, USA, 93–100. https://doi.org/10.1145/964442.964461

[8] Mary Goldberg, and Jonathan Pearlman. 2013. Best Practices for Team-Based Assistive Technology Design Courses. Annals of Biomedical Engineering. Vol. 41, Issue 2013. 1880–1888. https://doi.org/10.1007/s10439-013-0798-2

[9] Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. 2008. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (Assets '08). Association for Computing Machinery, New York, NY, USA, 73–80. https://doi.org/10.1145/1414471.1414487

[10] Stephanie Ludi, Matt Huenerfauth, Vicki Hanson, Nidhi Rajendra Palan, and Paula Conn. 2018. Teaching Inclusive Thinking to Undergraduate Students in Computing Programs. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 717–722.. DOI: https://doi.org/10.1145/3159450.3159512

[11] Meredith Ringel Morris. 2020. AI and accessibility. Communications of the. ACM 63, 6 (June 2020), 35–37. https://doi.org/10.1145/3356727

[12] Alannah Oleson, Amy J. Ko, Richard Ladner (Eds.) (2023). Teaching Accessible Computing. Self-Published. https://bookish.press/tac, retrieved 11/28/2023.

[13] PEAT website. Retrieved January 5, 2023 from https://www.peatworks.org/

[14] Ashraf Saad, Senior capstone design experiences for ABET accredited undergraduate electrical and computer engineering education. Proceedings 2007 IEEE SoutheastCon, Richmond, VA, USA, 2007, pp. 294-299, https://ieeexplore.ieee.org/abstract/document/4147435.

[15] Lawrence A. Scadden, Kurzweil reading machine: Evaluation of model one. Journal of Visual Impairment & Blindness 1978. Volume 72. Issue 10. 415-418.

[16] Teach Access website. Retrieved September 10, 2022 from http://www.teachaccess.org
[17] Kendra Walther and Richard E. Ladner. 2021. Broadening participation by teaching accessibility. Communications of the. ACM 64, 10 (October 2021), 19–21. https://doi.org/10.1145/3481356
[18] WCAG website. Retrieved November 6, 2022 from https://www.w3.org/WAI/standards-guidelines/wcag/
[19] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-Based Design: Concept, Principles and Examples. ACM Trans. Access. Comput. 3, 3, Article 9 (April 2011), 27 pages. https://dl.acm.org/doi/10.1145/1952383.1952384