Teaching Accessibility: A Design Exploration of Faculty Professional Development at Scale

Saba Kawas
The Information School | DUB Group
University of Washington
Seattle, Washington
skawas@uw.edu

Laura Vonessen
Paul G. Allen School of Computer Science | DUB Group
University of Washington
Seattle, Washington
laurav4@cs.washington.edu

Andrew J. Ko
The Information School | DUB Group
University of Washington
Seattle, Washington
ajko@uw.edu

ABSTRACT

Most CS students learn little about accessibility in higher education; this is partly because most CS faculty know little about accessibility. Unfortunately, higher education CS faculty lack a model of professional development for learning to teach new topics. Therefore, we investigated the feasibility of a "micro" professional development model for teaching accessibility in CS courses that could be used at scale. We conducted 18 semi-structured interviews with U.S. CS faculty, asking them to explore a prototype of a web-based professional development tool that linked accessibility topics to CS topics. We found that many organizational factors limited faculty's autonomy to integrate accessibility in many of their courses. We also found that individual values and knowledge constrained faculty's ability and willingness to both learn and integrate accessibility topics into their courses. However, many faculty expressed desire to teach accessibility in their courses if they had access to even basic accessibility content and materials to use in their courses.

CCS CONCEPTS

• **Social and professional topics → Computing education programs**; • **Human-centered computing** → *Accessibility theory, concepts and paradigms*;

KEYWORDS

Accessibility; professional development; higher education

ACM Reference Format:
Saba Kawas, Laura Vonessen, and Andrew J. Ko. 2019. Teaching Accessibility: A Design Exploration of Faculty Professional Development at Scale. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), February 27-March 2, 2019, Minneapolis, MN, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3287324.3287399

1 INTRODUCTION

Many companies want to provide accessible products and services, reaching people with a wide range of abilities. To achieve this, companies need software engineers to have knowledge of how to build accessible software. Unfortunately, most software designers and engineers are not taught about accessibility in courses [19], and so many software designers and engineers overlook the accessibility of software products and services.

Fortunately, many CS faculty are open to learning. A recent national survey of higher education CS faculty in the U.S. showed that faculty across most areas of CS are interested in learning how accessibility topics might be integrated into their specific courses [18]. Faculty report three things to teach accessibility part of CS courses: sufficient expertise on accessibility to teach these topics, concrete ideas about how to integrate accessibility into their specific sub-disciplines of CS, and time.

Some efforts have begun to provide these, but not in scalable ways. For example, the Teach Access consortium (teachaccess.org), including Facebook, Microsoft, Google, Adobe, and others, formed in 2016 to advocate for the teaching of accessibility in CS in higher education. It encourages faculty to teach more about accessibility, develops accessibility learning materials, has begun offering professional development workshops for faculty, and has offered small grants to individual faculty willing to update their course materials with accessibility content. And for the small number of earlier adopters, there is a lot of content already. For example, there are resources on accessibility in general [6], materials on accessibility in the context of CS from organizations such as AccessComputing [8] and TeachAccess, and a range of published research ideas on how to integrate accessibility into specific subjects, such as software engineering [9–11], HCI courses [16, 17], and web development [4, 8].

Unfortunately, finding scalable ways to encourage these integrations, and scalable ways to train faculty on the pedagogy required to succeed at teaching them [2, 14, 15], is not straightforward [13, 20]. This problem is not specific to teaching accessibility in CS. In K-12 education, for example, teachers are required to pursue professional development, including evening, weekend, and summer classes to learn new teaching methods, content areas, and techniques for managing classrooms [3]. These mechanisms scale teacher learning by incentivizing or even requiring teachers to improve their practice and learn new content knowledge. And with good reason: high quality professional development can have measurable improvements on both teacher and student learning, and can lead to concrete change in what and how teachers teach [12].

Most higher education faculty have no such external incentive to learn. This means that most change in higher education must come

SIGCSE '19, February 27-March 2, 2019, Minneapolis, MN, USA

ACM ISBN 978-1-4503-5890-3/19/02…$15.00
https://doi.org/10.1145/3287324.3287399

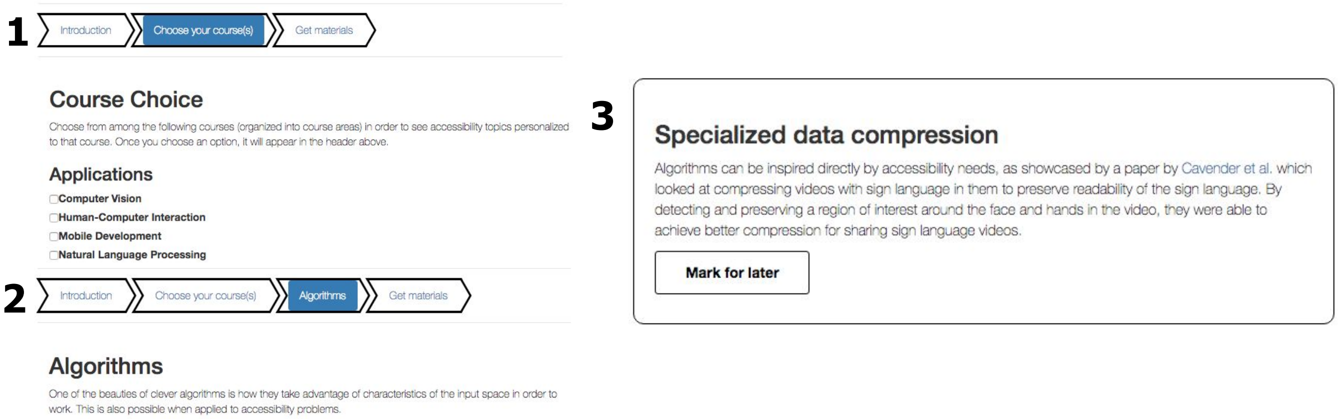

Figure 1: The three stages of interaction in screenshots of our accessibility micro-PD prototype: 1) choosing courses from broad areas, 2) selecting an accessibility learning objective, and 3) learning the accessibility concept and how to teach it.

in small, incremental, well-timed, and individually-led interventions [5]. Administrators have found that incentives and resources are key, but only work when faculty find time to learn and change what they teach [1]. This suggests that even if CS faculty are willing to learn and teach about accessibility, they need resources that are closely integrated with what they already teach, easy to learn, and require no major time commitments. We ask the following research question: how might we provide such professional development about accessibility to higher education CS faculty, in the absence of time and incentives?

In this paper, we explore the feasibility of an idea we call “micro” professional development (micro-PD): a personalized, integrated, and low-commitment approach to teaching accessibility. We describe the micro-PD concept, the content it would require, and the results of an evaluation study of an early prototype of micro-PD prototype at the 2018 SIGCSE Technical Symposium.

2 MICRO PROFESSIONAL DEVELOPMENT

Our vision of accessibility micro-PD is for it to reduce the amount of time it takes for CS faculty to learn about accessibility and integrate concepts into their courses. Our approach was to create a new genre of web-based instruction, in which faculty would visit a website, select courses relevant to their CS teaching expertise, learn accessibility concepts that are specifically relevant to the courses they teach, and then obtain relevant materials to adapt and integrate into their course. The key idea underlying this approach is to create a *mapping* from CS learning objectives to accessibility learning objectives and offer materials that teach these specific links. This approach deviates from conventional forms of teacher professional development, which require faculty to sign up for entire classes or complete online training [3].

Figure 1 portrays our prototype for micro-PD, showing three stages of interaction. The first stage is to identify which learning objectives a teacher has expertise to teach. To streamline this, our prototype used courses as collections of learning objectives, asking faculty to identify which courses they have expertise to teach. While showing individual learning objectives would provide the most granularity, it would pose a high burden, since there are hundreds of learning objectives in CS curricular frameworks like the ACM Computing Curricula. Figure 1 shows the first few of the 19 courses in our list.

The second stage in the prototype is to identify accessibility learning objectives that are relevant to the selected CS learning objectives. Table 1 shows examples of CS learning objectives in specific CS courses that can be mapped to accessibility learning objects for different courses they could teach. For example, in NLP, one can talk about applications of NLP to problems of speech and hearing, and even pose challenges such as recognizing the speech of deaf speakers. Or, in data structures courses, one can use problems that involve representation and translation of Braille characters to Unicode characters.

CS faculty should be expected to know or identify these relevant links independently. We identified them by mining AccessComputing’s knowledge base and W3C standards. We also found links from the other direction, examining CS learning objectives from the ACM Computing Curriculum Guidelines and using our accessibility knowledge to identify links that weren’t clear in other sources. We interviewed two Computer Science accessibility faculty experts for any additional links and asked them to review the micro-PD content. This resulted in 72 accessibility learning objectives mapped to CS learning objectives.

Through this process, we discovered four types of links between CS and accessibility:

- **Equivalent objectives**. Some accessibility concepts are *equivalent* to CS learning objectives. For example, HCI is about designing computer systems while taking the people who will use them into account; these people will have varying abilities.
- **Accessibility as examples**. Some accessibility topics can be used as examples to illustrate CS concepts. For example, a database course might discuss data modeling problems through disability, which is a good example of a complex, nuanced data type that schemas often oversimplify (is disability binary, enumerated?).

Table 1: Sample of Mapping Accessibility Topics to Learning Objectives in CS Courses

Computer Science Course	Learning Objective Example
Computer Vision	Students should know how (1) optical character recognition (OCR) can recover textual information from scanned text, which is useful for blind and people with low vision, and that (2) object recognition can help automatically generate ALT text for existing images on the Internet.
Programming Languages	Students should know that people who are blind can read and write code using screen readers because screen readers navigate hierarchies, and program's abstract syntax trees are inherently hierarchical.
Networks	Students should know that many communication technologies were motivated by the needs of people with disabilities. For example, the modem was invented to transmit text over phone lines for people who are deaf [7].
Natural Language Processing (NLP)	Students should know that the quality of speech recognition algorithms depends on the quality of the data used to train them, and that such data should include people who are deaf, hard of hearing, or include speech impediments.
Visualization	Students should know that nearly 8.5% of people are color blind, and therefore color choices in visualizations can impact what is perceptible.
Robotics	Students should know that many of the primary applications of robotics are in assistive contexts for people with motor impairments, such as robotic walkers, intelligent artificial limbs, and autonomous vehicles.
Data structures	Students should know that text can be represented in many different ways, including popular encodings such as ASCII and Unicode, but also Braille and sign language.
Algorithms	Students should know that accessibility poses many grand challenges in algorithms, such as efficiently transmitting high frame rate of sign language video and finding walking routes for people with mobility issues.
Introduction to programming	Students should know that the syntax of a language is different than its semantics, and can learn this by learning how programmers who are blind read and comprehend code independent of its syntax.
Software engineering	Students should know that accessibility frameworks are an exemplar of *code reuse*, since they can take care of many of the accessibility concerns for you when used properly. For example, default buttons in an Android application can expose information to the android screen reader without you having to write your own screen reader.
Human Computer Interaction	Students should know what accessibility is, how it relates to user interface design, how to design and implement accessible interfaces, and how different disabilities can be supported through universal and ability-based design principles.

- **Accessibility as context**. Some accessibility knowledge serves as historical context for CS. For example, text messaging has its origins in TTY, a way deaf people used to transmit text over phone lines, which was made possible by a modem that deaf physicist Robert H. Weitbrecht invented [7].
- **Accessibility as motivating problems**. Accessibility problems can prompt interesting applications of core CS topics. For example, people who are deaf can benefit greatly from real-time captioning, which is a grand challenge for real-time natural language processing.

After stage two identifies relevant accessibility learning objectives, the last step is teaching the instructor the accessibility concepts and providing examples for how to integrate them into the teaching of the CS learning objective. Figure 1 shows an example of a mapping that results from choosing Algorithms course, linking the learning objective of understanding data compression to the accessibility problem of achieving high fidelity video for streaming sign language. Our vision is for the micro-PD to include all the links we identified and course materials such as slides to teach the concept to students in class. For this paper, our prototype contained 36 mappings distributed across 19 courses, but no course materials.

3 FEASIBILITY EVALUATION

To investigate whether the micro-PD content we described in the previous section is sufficient to support faculty learning and integration of accessibility concepts. We designed a feasibility study in which we would simulate a cold outreach to random CS faculty and observe their response to our prototype and its content. While distributing emails would have most closely mimicked our intended deployment at scale, this would have made it difficult to richly capture faculty's in situ responses. Therefore, we instead planned a study at the 2018 ACM SIGCSE Technical Symposium on Computer Science Education, where many CS faculty go to learn about new insights from the research on and practice of CS teaching and learning. Our approach was to find faculty during breaks and meals, ask them for 10-15 minutes of their time to learn about teaching accessibility, and elicit their reactions. In the rest of this section, we detail our method and describe our results.

3.1 Method

Of the 1,753 attendees at SIGCSE 2018, we approached 40, either between sessions, or at the end of three sessions about accessibility,

Table 2: Participant demographics

Gender	Male (12), Female (6)
Department	Computer Science (14), Department of Electrical and Computer Engineering (2) College of Cyber (1), Creative and Applied Computing/Math (1)
Title (some w/ multiple roles)	Professor (3), Associate Professor (6), Assistant Professor (3), Instructional Assistant Professor (1), Instructor (3), Associate Dean (1), Chair (1), Research Scientist (2), Sr. Software engineer (1)
Years of teaching experience	1-5 (3), 6-10 (5), 11-15 (4), 16-20 (3), 21-25 (2), 26+ (1)
Knowledge of accessibility (self-reported)	Not knowledgeable (4), Some knowledge (10), Knowledgeable (4), Expert (0)
Main Role	Teaching (15), Research (9), Administration (6), Software development (1)
Expertise areas (some with multiple)	HCI (4), CS education (3), Software engineering (2), Web development (2), AI and machine learning (2), Networks (2), Intro programming (2), Systems (2), Programming languages(1), Algorithms (1), Cybersecurity (1), GIS (1), Project management (1), Computational Science and Engineering (1), Computational optimization (1), Databases (1), Computer organization (1), Bio-informatics (1), Operating systems (1), Honey bee technology (1), Embedded systems (1), Complexity (1)
Class size	1-10 (1), 11-20 (2), 21-30 (8), 31-40 (2), 80 (1), 100 (1), 120 (1), 200 (1), 500 (1)
Course level	Lower-level undergraduate (9), Upper-level undergraduate (7), Masters (1), Ph.D. (1)

which attracted attendees interested in learning to teach about accessibility and teaching inclusively. Of the 40, 12 declined (due to lack of time, interest, or perceived relevance of accessibility), or did not attend our scheduled time, and 10 did not meet our inclusion criteria, having never served as a CS instructor in higher education. This left 18 faculty participants, of which 6 were women. We stopped recruiting when the issues that faculty raised were no longer new, as our goal was to identify the range of reactions, but not their prevalence.

Faculty had 2 to 49 years of teaching experience (median 12) and ranged from lecturers to full professors, including an associate dean and a department chair. Their main duties included teaching (15), research (9), administration (6), and software development (1). Only 4 reported being knowledgeable about accessibility. Faculty had expertise in 22 distinct CS areas, from theoretical areas such as complexity theory to applied areas such as web development and software engineering. Class sizes and levels varied (11 with 30 students or less, four with 100 or more). The resulting sample therefore captured a diversity of contexts, topics, and expertise of CS faculty. Table 2 has detailed participants' information.

When an instructor agreed to participate, we began by asking them to think of the next course they are or would prepare for in the next couple of months. We had them imagine they just received an email from their department chair encouraging them to take a look at our micro-PD web-based tool to decide whether it would be useful to them. We encouraged faculty to start with that specific course in mind, but if our prototype did not include that course, to explore the course closest to their expertise. While they used the micro-PD site, we audio- and video-recorded their interactions and prompted them to think aloud about their reactions to the content, and asking clarifying questions. If faculty were silent for too long, we asked them what they were thinking, or to elaborate on interactions we could not interpret. After they were done exploring, we performed a semi-structured interview asking about the potential barriers and support needed to use this tool, as well as to explore issues that came up during the think-aloud while using the micro-PD prototype. We also asked about the broader issues of motivation, expectations, and constraints that might prevent them from using this tool and any ideas to improve it. After the interview was complete, we had instructors fill out a demographic survey. Interactions with faculty ranged from 13 to 26 minutes.

3.2 Results

To analyze the transcripts, field notes, and video, primary and secondary researchers used deductive coding with continuous comparative analysis, developing a codebook to identify feasibility issues with our accessibility micro-PD. The researchers independently coded statements from the transcripts against these codes and then discussed disagreements with the themes to further refine them. We followed Hammer and Berland's [5] best practices on qualitative coding, not treating the coding results as data but as an organization of claims about the data. Therefore, we focused more on deeply discussing and resolving disagreements between codes than achieving inter-rater reliability of coding.

Overall, faculty's reaction to the concept of micro-PD was positive. They liked the idea of having access to teaching materials that would be easily integrated into their current courses. That said, there were numerous barriers they saw before such a vision would be viable. At the high level, these ranged from extrinsic factors such as institutional context and policy, the curricular factors that constrained course changes, individual faculty constraints, and perspectives on the micro-PD content itself.

3.2.1 Organizational constraints. Several of the reactions that faculty expressed concerned organizational factors that constrained or influenced their course preparation decisions. For example, as faculty viewed the content, one point of tension some faculty surfaced were the ACM curricula recommendations, and how they constrained adoption. There were varying degrees to which faculty felt the standard constrained them or was relevant when preparing their courses:

> When we revise our curriculum, we go to the ACM guidelines and make sure that we're matching that. When I'm designing a course I don't do that, but when you're revising a curriculum you do.

In at least some cases, this distinction was because courses simply were not included in the ACM curricula recommendations, such as "Intelligent game-based learning environments" or "Accessibility."

The lack of visibility of accessibility in the ACM curricula (only mentioned in the two knowledge areas HCI and Social Issues and Professional Practice [19]), was also mentioned as problematic. In particular, this instructor was discussing her technique of mentioning accessibility at a high-level in introductory courses and whether this was part of the ACM recommendations:

> There's ACM guidelines that sort of tell you what you should be covering... I've not looked at those guidelines in a while, but...I doubt that [it is].

Even when faculty could see how accessibility fit, many instructors commented that courses were already full of required topics, with no room to add on anything extra without having to take something else out:

> [This material] would have to embed without adding extra time because all of us have courses that are already crammed in there.

Even when faculty saw ways of integrating accessibility content without adding content, and they personally wanted to integrate it, many shared that department policies constrained them. For example, some of the content suggested discussing more accessible languages such as the Quorum programming language [17]:

> We use C++ and I don 't have any control over...that

Other faculty perceived more intellectual freedom in their choices of what and how to teach CS learning objectives:

> There's nothing at an institutional level—I'm not being handcuffed by..some kind of departmental restrictions on what I teach in the course.

Some faculty felt they had freedom but feared they would not have support. In one case, an accessibility topic was even a required topic in a course, but the instructor felt he did not have enough support in how to teach it and so did not. In some cases, department, university, and student values interacted with faculty's willingness to integrate content:

> The...university has an objective what that class needs to cover, but then the students [also] have an objective of what they want to get out of the class ... To what extent, of that limited time, would it be appropriate to spend discussing these [accessibility] topics?

Some instructors mentioned that being able to show students concrete examples that companies are interested in accessibility can be a way of catching their interest.

> If I were to show this and say, "Hey, you know even Microsoft thinks this is important," and they see the little logo on that...they can get excited about that.

3.2.2 Individual constraints. While faculty viewed organizational and course constraints as unavoidable, they also lamented their own personal values, knowledge and resources constraints.

Not surprisingly, many of the faculty said that they were far too limited in their own time to prepare for teaching, research, and other academic and service duties to have time to innovate in teaching:

> Time would be an issue ... you're fighting fires all the time.

More surprisingly, however, was that some faculty were simply worried that they would forget that such a resource exists or never find it. Some faculty suggested needing multiple reminders about the content or having multiple broadcasts on the SIGCSE mailing list for lists of resources. One instructor mentioned that while in the process of preparing for an algorithms course it wouldn't occur to him to look for a resource on accessibility, but if it were part of a collected list of resources on teaching algorithms he would be more likely to stumble across it.

While most reported needing time, and most faculty reported having little knowledge about accessibility, many were enthusiastic about the potential of the micro-PD tool to provide them sufficient expertise. One faculty, for example, who had only 5 years of teaching experience, and no knowledge of accessibility, started generating ideas for an accessibility assignment after learning some of the basic concepts in accessibility:

> Our CS 2 is where we do like object-oriented stuff, so you could say, "Here's a generic compression thing," and then, "Okay, now what about specialized kinds of compressions?" And then also highlight [to] them that accessibility portion, too... Use it as an introduction to a new topic slash introduction to an assignment for them...I think it would make a cool assignment.

The value faculty saw in teaching accessibility varied. For example, one instructor was interested in learning more about accessibility for a course which included web development topics, but skeptical about its relevance to the operating systems course he taught. Others were more enthusiastic or even already taught accessibility courses or were in the process of putting one together. When faculty did not see or agree with a link between their topic and accessibility, some saw this as a sign that accessibility wasn't relevant to their version of the course, while others enjoyed the serendipity of seeing new ideas.

> I hadn't thought about this... I would definitely incorporate it...that's what I use these things for, sparking ideas.

3.2.3 Reactions to content. One clear pattern that emerged when faculty engaged in the content was that they would likely not use it without modification. Some faculty mentioned that they often get materials from their faculty colleagues, and so getting recommendations with links to examples from our micro-PD would be ideal. All faculty said that even when borrowing content, they would adapt it to their own use. In reference to the content in our micro-PD tool:

> [Slides] would have to be customized a little bit, but that would be fine.

Some faculty noted that many courses with the same titles have dramatically different content, and so organizing accessibility topics by courses often led to misalignments. For example, a course on programming languages might be about a specific programming language, language theory in general, or it could be about the design of new programming languages. Some faculty recommended adding faceted browsing features to help them narrow down to the set of learning objectives most relevant to their course:

> You might consider ... helping them filter into what their area is to get that set of things versus trying to come at the course level where you may have missed something.

Faculty also wanted significantly more motivation for why accessibility was relevant to the course, and why it was relevant to students:

> How do I naturally bring up those topics so it doesn't feel like it's forced or artificial?

Most instructors felt a need for more information on where these accessibility topics came from, although not always for the same reasons. Some wanted to verify licensing issues, asking about Creative Commons licenses for the content. Others wanted to verify that the content was developed by a legitimate, credible source of expertise. One wanted to be able to learn more about accessibility topics to be prepared for student questions, and a few wanted to share links with students so they could find out more on their own.

> It may be something that they fall back on... Say they present the slides, a student asks an annoying but actually really good question [and] I'm stumped—I'm going to go back and dig into the deeper stuff.

Nearly all faculty desired for course materials provided by the tool to go beyond just slides. Faculty mentioned many possibilities, such as activities, assignments, homework or exam problems, projects, videos, simulations, examples, and data sets.

Some faculty expressed confusion about the audience of the tool. For example, one at first interpreted the content being presented to the instructor as a handout for students, and others wanted to be able to direct students directly to the tool for flipped classroom activities.

> Since I usually use a flipped classroom, I would like to know if this would be suitable for the students to view on their own and then if there are activities that I can use in the classroom.

Other faculty thought this tool would be a way to make their courses accessible for students with disabilities. This repeats a fundamental confusion that Shinohara et al. also discussed [18], which one instructor articulated in the following way:

> Maybe at the high level there's two branches of, "How do you make your own content accessible?" and, "How do you teach accessibility to others?" Having that be clear—that's where I got confused, but ... both of those things are of interest.

4 DISCUSSION

While our data show that the faculty in our sample viewed the idea of micro-PD as promising and valuable, many complex factors constrained their willingness and ability to change course content. These exist at many different levels: curricular frameworks narrow their choices, departmental values create fear of change, and faculty's own busy schedules limit the time they are willing to engage with even small amounts of professional development. We also observed divergent reactions to our preliminary content, with some faculty finding it highly useful and informative, while others could not imagine how to adapt it to their needs.

Some limitations in our prototype and evaluation hinder our ability to interpret the results. One factor is in who we were able to recruit. SIGCSE attendees may be more open to learning about teaching and changing their instruction. Some of the faculty we approached declined to participate due to lack of interest or doubt that accessibility topics could be relevant to the courses they teach. Additionally, faculty we recruited at the accessibility sessions were more motivated about accessibility. Another factor is that some courses had more accessibility content than others. We also didn't evaluate the tool at scale and whether faculty would be motivated to use it. While one of the main micro-PD goals is to provide accessibility materials that are delivered on-line in an organized and small chunks to make it digestible and easy to access by faculty who don't have time. We did not quite mimic someone receiving an email with a link to the content, so some faculty may not have explored independently as much as they did in our face-to-face interactions.

Although there are limitations on our interpretation, our results do show the range of reactions to the concept of micro-PD for teaching accessibility, and the flexibility it would require to serve this range. Micro-PD would need both more content and a range of course material formats for integration to support the diversity of teaching styles that exist. Additionally, because most faculty were likely more motivated to engage because of the conference and the face-to-face interactions in our study, the bar for polished, highly personalized, and highly adaptable web-based content is likely higher than our results might suggest.

These interpretations suggest several avenues for future research. First, we need further investigation into precisely how to structure and present content so it is easy to integrate and appropriate. A more detailed case study of faculty use of content could elicit these requirements. Second, with enough content ready for adoption, a pilot deployment to CS faculty would also reveal to what extent the findings in this study would also occur in the even more complex reality of daily faculty life.

While more research is needed, our data does show that there are already faculty who are motivated to teach accessibility, and would find even basic materials such as the ones we shared more than adequate to get started. If we can come together as a community to develop this content, building a community of accessibility experts to fully express the micro-PD vision in this paper, we may be able to reach an order of magnitude more CS faculty than currently teach accessibility. And with this effort, we can reach more students, and therefore more end users, with more accessible software.

ACKNOWLEDGMENTS

The authors would like to thank all the faculty who participated in our evaluative study at SIGCSE 2018. The project was approved by the Institutional Review Board IRB ID STUDY00004124. This work is supported by National Science Foundation Grants 1539179, 1314399, 1240786, 1153625, and Microsoft, Google, Adobe.

REFERENCES

[1] M. Besterfield-Sacre, M. B. Monica F Cox, K. Beddoes, and J. Zhu. 2014. Changing engineering education: Views of US faculty, chairs, and deans. *Journal of Engineering Education* 103, 2 (2014), 193–219. https://doi.org/10.1002/jee.20043

[2] K. E. Bigelow. 2012. Designing for success: Developing engineers who consider universal design principles. *Journal of Postsecondary Education and Disability* 25, 3 (2012), 211–225.

[3] H. Borko. 2004. Professional development and teacher learning: Mapping the terrain. *Educational Researcher* 33, 8 (2004), 3–15. https://doi.org/10.3102/0013189X033008003

[4] J.Rosmaita Brian. [n. d.]. Accessibility first!: a new approach to web design. ACM SIGCSE Bulletin.

[5] W. Dennis. 2001. Teaching science in higher education: Faculty professional development and barriers to change. *School Science and Mathematics* 101, 5 (2001), 246–257. https://doi.org/10.1111/j.1949-8594.2001.tb18027.x

[6] E. A. Draffan, M. Wald, K. Dickens, G. Zimmermann, S. Kelle, K. Miesenberger, and A. Petz. 2015. Stepwise approach to accessible MOOC development. *Studies in Health Technology and Informatics* 217 (2015), 227–234.

[7] ITU News-Accessibility for all. June,2011. How the deaf developed a phone of their own. International Telecommunication Union News. https://www.itu.int/net/itunews/issues/2011/05/36.aspx Last accessed 16 April 2018.

[8] A. J. Ko, R. E. Ladner, Andrew J Ko, and Richard E Ladner. 2016. AccessComputing Promotes Teaching Accessibility. *ACM Inroads* 7, 4 (2016), 65–68. https://doi.org/10.1145/2968453

[9] Stephanie Ludi. 2002. Access for everyone: introducing accessibility issues to students in Internet programming courses. https://doi.org/10.1109/FIE.2002.1158617

[10] Stephanie Ludi. 2007. Introducing accessibility requirements through external stakeholder utilization in an undergraduate requirements engineering course.. In *Proceedings of the 29th International Conference on Software Engineering*. 736–743. https://doi.org/10.1109/ICSE.2007.46

[11] Israel Martin-Escalona, Francisco Barcelo-Arroyo, and Enrica Zola. 2013. The introduction of a topic on accessibility in several engineering degrees.. In *Global Engineering Education Conference (EDUCON)*. 656–663. https://doi.org/10.1109/EduCon.2013.6530177

[12] C. Mouza. 2009. Does research-based professional development make a difference? A longitudinal investigation of teacher learning in technology integration. *Teachers College Record* 111, 5 (2009), 1195–1241.

[13] Bohma Paul. [n. d.]. Teaching accessibility and design-for-all in the information and communication technology curriculum: Three case studies of universities in the United States, England, and Austria.

[14] Cynthia Putnam, Maria Dahman, Emma Rose, Jinghui Cheng, and Glenn Bradford. 2015. Teaching Accessibility, Learning Empathy.. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. 333–334. https://doi.org/10.1145/2700648.2811365

[15] Cynthia Putnam, Maria Dahman, Emma Rose, Jinghui Cheng, and Glenn Bradford. 2016. https://doi.org/10.1145/2831424

[16] Kristen Shinohara. 2016. Cynthia L Bennett, and Jacob O Wobbrock.. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. 229–237. https://doi.org/10.1145/2982142.2982158

[17] Kristen Shinohara. 2017. Teaching Accessibility in a Technology Design Course.. In *12th International Conference on Computer Supported Collaborative Learning (CSCL)*. International Society of the Learning Sciences, Philadelphia, PA.

[18] Kristen Shinohara, Saba Kawas, Andrew J. Ko, and Richard E. Ladner. 2018.

[19] C. Velasco et al. 2004. IDCnet: inclusive design curriculum network – first results. https://doi.org/10.1007/978-3-540-27817-7{_}16

[20] Annalu Waller. 2009. Vicki L Hanson, and David Sloan.. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*. 155–162.