Editor: **Brittany Johnson**
George Mason University
johnsonb@gmj.edu

Editor: **Tim Menzies**
North Carolina State University
tim@menzies.us

The Power of Positionality—Why Accessibility? An Interview With Kevin Moran and Arun Krishnavajjala

Brittany Johnson and Tim Menzies

From the Editors

Software should be accessible to everyone. Otherwise, is it truly serving its purpose? Everyone, including people with disabilities, knows that software is useless when it can't be used. In this "SE and Ethics" column, we interview Dr. Kevin Moran, University of Central Florida, and his student Arun Krishnavajjala about how to make software more accessible to a wider audience. More specifically, we explore proactive considerations for a variety of disabilities, starting before any code is ever written.

This column collects news and views on issues of software engineering and ethics. Got something to say on that topic? If so, e-mail a one-paragraph synopsis to timm@ieee.org or johnsonb@gmu.edu (subject line: "SE Ethics Idea: [Your Idea]"). If that looks interesting, we'll ask you to submit a 1,000–3,000-word article (where each graph, table, or figure is worth 250 words) for review for *IEEE Software.—Tim Menzies and Brittany Johnson*

WHAT IF WE could create something where developers can make their applications accessible from the start, where they can still innovate and make cool apps but also make them accessible at the same time?

This is the shared vision of third-year Ph.D. student Arun Krishnavajjala and his advisor Dr. Kevin Moran, who are working together to provide automated tool support for designing accessible mobile applications.[1]

Digital Object Identifier 10.1109/MS.2024.3360650
Date of current version: 11 April 2024

The Power of Positionality—Why Accessibility?

For both researchers, these endeavors have personal motivations. Dr. Moran knows firsthand the frustrations that can come with technology not meeting all users' needs: "I have a family member who is in a wheelchair and is motor impaired. I actually was not really planning to get involved in this area of research until the pandemic happened and our primary means of communicating with her was through software. It turns out that a lot of it does not work really

ABOUT DR. KEVIN MORAN AND ARUN KRISHNAVAJJALA

Dr. Kevin Moran is an assistant professor in the Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA. He is affiliated with the Cyber Security and Privacy Cluster and directs the Software Automation, Generation, and Engineering Lab. For more information, see https://www.kpmoran.com/.

Dr. Kevin Moran.

Arun Krishnavajjala is a Ph.D. student in the Department of Computer Science, George Mason University, Fairfax, VA 22030 USA. He works in the Software Automation, Generation, and Engineering Lab, where he focuses on using machine learning and computer vision techniques to create developer-facing tools that enhance the software development workflow, bridging the gap between innovation and practical implementation. For more information, see https://www.arunkv.com/.

Arun Krishnavajjala.

well for someone who is operating a device using a switch or some other assistive mechanism."

For Arun, the motivation also came from his family. It was his mother and her work with people with disabilities that contributed to his desire to do this work. "My mom teaches autistic elementary students. So, I kind of see it on a firsthand level," he says. Fortunately for Arun and Dr. Moran, they have the power and positionality as software engineering researchers to contribute solutions.

Toward Proactive Accessibility Support

While their budding interest in software accessibility is new for them both, the space of developer tools for software accessibility is not. In fact, prior efforts set the foundation for the initiatives that Arun and Dr. Moran have been working on.[2,3] But they noticed a gap with respect to tool support for mobile app design and development. In particular, they wanted to support proactive considerations for a variety of disabilities, before any code is ever written.

"This is a fairly new area of work in software engineering research. There's been a couple of groups that have been doing really good work in this space already … but one thing we noticed when surveying the literature is that the focus has been disproportionately on a smaller set of users with certain types of disabilities," states Dr. Moran.

Both researchers noted that when doing this kind of work, it's both useful and important to think about how the software or application can be inclusive of all demographics of users. Fortunately, there are foundations in place to support this kind of consideration in the form of guidelines.[4,5,6] Many of these guidelines are targeted toward supporting users of varying backgrounds and abilities to ensure they can use software as easily as the general population.

For example, one important guideline is about how expanding sections (such as popup menus) affect users with motor impairments using mobile app (see Figure 1). Figure 1(a) shows a screen that has an explicit button to close an expanding section, which makes it easy for users who might interact with an app via an assistive mechanism, such as a physical switch. Users who interact with a user interface (UI) through a switch rely on a scanning mechanism that iteratively highlights different UI elements, allowing them to tap the external switch to "tap" on the highlighted element. However, the screen in Figure 1(b) does not include a way to easily close a menu, which can make it difficult for motor-impaired users to navigate since the UI element scanning mechanism cannot highlight a button to close the menu.

Dr. Moran says, "You could argue that there's a really long tail of different accessibility features that could be included in software, but there's a pretty established set of guidelines that go a long way to support considering accessibility issues for a variety of different demographics."

Arun adds to this, noting, "These guidelines have been around for a really long time. It's only recently that technology has caught up to being able to automate these things. It's surprising to me how little work there is on these guidelines that have existed for so long." While he and Dr. Moran agree that this is unfortunate, they also emphasize the opportunity it presents for them and other researchers to fill the gap.

Arun and Dr. Moran remarked that much of the recent research that we have seen at the intersection of software engineering and accessibility has been fueled by recent breakthroughs in artificial intelligence (AI) and deep learning. For instance, recent advancements in computer vision-based screen understanding, capable of reverse engineering various properties of UIs,[6] have allowed for accessibility techniques that infer information from UIs and then use this information to improve screen accessibility.[7,8] For example, recently, researchers have created techniques that adapt approaches for neural image captioning to automatically generate descriptions, known as *alt text*, for UI icons that make it easier for users with visual impairments to navigate software. However, both Dr. Moran and Arun note that we are just scratching the surface of the tools and techniques that we can create by leveraging these advancements in AI.

Dr. Moran notes, "In the end, you want a seamless development process. From the get-go, you want a designer to envision the app and adjust the design to make sure they fall within accessibility standards. On the implementation side, if you're not sure how to increase contrast of text or create an accessible menu, having a code autocompletion engine focused on these kinds of contributions could be really useful. On the debugging and maintenance side, if something does sneak through automated support, [it is important] to understand and rapidly fix any accessibility issue that arises."

In fact, the opportunities to provide support beyond the detection of accessibility issues are vast. This includes leaning into our software engineering foundations to ensure adequate documentation and transparency for end users to know that the applications they are using are in fact accessible and in what ways.

Advancing Accessibility— A Call to Action

Foundations have been laid, from guidelines to automated checkers, for accessibility considerations and support. So, what else do we need to realize effective tool support for software accessibility in practice?

For Arun, it's data. In fact, much of his current effort has focused on finding and labeling datasets, given the lack of representative robust datasets on disabilities that need to be considered. "There's so many niche disabilities to account for. If we just say 'visual impairments,' what does that mean? That could mean partial blindness or just complete loss of vision." Dr. Moran agrees, adding, "We haven't been able to work directly with our target audience. I think a key part of any accessibility research is that you really do need to be working with the populations

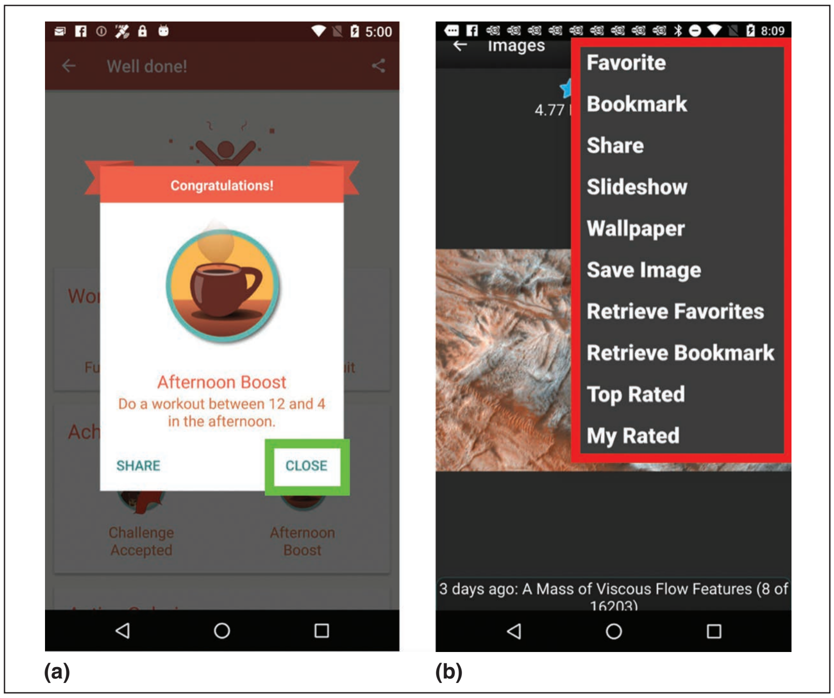
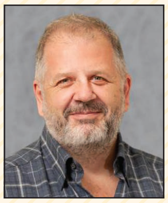


FIGURE 1. Screens (a) with and (b) without a button to close an expanding section.

ABOUT THE AUTHORS

BRITTANY JOHNSON is an assistant professor of computer science at George Mason University, Fairfax, VA 22030 USA. Contact her at johnsonb@gmu.edu.



TIM MENZIES is a full professor at North Carolina State University, Raleigh, NC 27606 USA. Contact him at timm@ieee.org.

affected. You don't want to prescribe something you think is an issue to them." He also acknowledges that while this is an important consideration, it is challenging in practice. "Development of these things is really important, but at the same time, it's really challenging, especially when you're in computer science, to find those kinds of partnerships."

He and Arun also mention the role of the government in this effort. Dr. Moran states, "The typical government relations we know about are [agencies like] the NSF. But I think there's potential for broader programs that are able to share anonymized data or connect us with the relevant communities that could facilitate this kind of work."

With all this in mind, he and Arun caution those interested in working on research in this space. They both emphasize the impact software engineers can have and the importance of knowing (and leaning into) our strengths while acknowledging what we can and can't do with them. "Don't try to solve problems you can't relate to. I cannot personally myself relate to accessibility concerns, but I relate to the fact that I am a software engineer. I'm able to provide input from my own experience to make better tools," Arun says.

Dr. Moran echoes and augments this sentiment, stating, "Broadly consider the impact of the job you're doing. A lot of work in software engineering is done behind a computer screen, but that doesn't lessen the impact." He encourages researchers to "think about the things you're passionate about as a researcher, and see how you can aim your research to make an impact on those things. Sometimes taking that leap can be really rewarding."

References

1. A. K. Vajjala, S. M. H. Mansur, J. Jose, and K. Moran, "MotorEase: Automated detection of motor impairment accessibility issues in mobile app UIs," in *Proc. Int. Conf. Softw. Eng.*, Lisbon, Portugal, to be published.
2. A. Alshayban, I. Ahmed, and S. Malek, "Accessibility issues in Android apps: State of affairs, sentiments, and ways forward," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, Jun. 2020, pp. 1323–1334, doi: 10.1145/3377811.3380392.
3. C. Vendome et al., "Can everyone use my app? An empirical study on accessibility in android apps," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, 2019, doi: 10.1109/ICSME.2019.00014.
4. "Build accessible apps." Developers. Accessed: Jan. 5, 2024. [Online]. Available: https://developer.android.com/guide/topics/ui/accessibility
5. "Accessibility." Apple. Accessed: Jan. 5, 2024. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/Accessibility
6. "Developers." Digital Accessibility. Accessed: Jan. 5, 2024. [Online]. Available: https://accessibility.huit.harvard.edu/developers
7. J. Wu, X. Zhang, J. Nichols, and J. P. Bigham, "Screen parsing: Towards reverse engineering of UI models from screenshots," in *Proc. 34th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 470–483, doi: 10.1145/3472749.3474763.
8. J. Chen et al., "Unblind your apps: Predicting natural-language labels for mobile GUI components by deep learning," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng. (ICSE)*, 2020, pp. 322–334, doi: 10.1145/3377811.3380327.
9. F. Mehralian, N. Salehnamadi, and S. Malek, "Data-driven accessibility repair revisited: On the effectiveness of generating labels for icons in Android apps," in *Proc. 29th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng. (ESEC/FSE)*, 2021, pp. 107–118, doi: 10.1145/3468264.3468604.