Brief Article

The Author

August 19, 2021

1 Introduction

2 Related Work

Effects of a hybrid interface and multimodal features for learning how to program [10, 6].
Benefits of a textual versus graphical interfaces for learning how to program [1].
Predicting user frustration and measuring perceived cognitive load [3, 7].
User evaluation methods and the impact of think-aloud on user testing [8, 2, 4].
Task analysis of the challenges of shell programming [5].
Optimizing visual and textual information presented on search user interfaces [9].

3 Study Setup

We conducted a remotely moderated, within subjects user study with X number of participants who had an average of X number of years of programming experience and regularly used Kubernetes in their work. Participants completed a Kubernetes task using the kubectl CLI and the OpenShift console. The order in which participants used the tools was counter balanced to mitigate learning bias between sessions, with half of the participants using the kubectl CLI first and the other half using the OpenShift console first []. We employed a classic "Think Aloud" user evaluation method during the sessions to avoid the effects of reactivity []. Video and audio from the sessions were recorded for later transcription and analysis.

Before the study sessions, participants filled out a background survey to provide information about their programming experience, use of Kubernetes, and familiarity with the two tools. After completing the study, participants filled out a follow up survey and ranked the tools based on their preference. In the follow up survey, participants were also asked to rate the tools based on their perceived cognitive load using the NASA TLX scale []. To give us a better understanding of preference, participants also rated the quality of each tool on a five-point Likert scale and were asked to give reasons for why they either liked or

didn't like using each tool. Participants were also asked to share their recommendations for how each tool could be improved.

We conducted pilot testing with two participants to identify any issues in our study design. One participant identified a misleading error in one of the task instructions so we rewrote it. Initially, we had only provided task hints for the kubectl CLI, and one participant realized that they could not apply these hints to the OpenShift console, so we also included task hints for the OpenShift console. We found that both participants were able to complete the task using each tool in under 30 minutes.

The goal of the Kubernetes task is to create an application in a namespace, find the deployment pods that have the string "magic key" in their logs, delete the application, and then delete the namespace. This task was chosen for our study because it is simple enough to be accomplished by a Kubernetes novice in multiple steps or by an expert in fewer steps and it requires participants to perform search and navigation using each tool. Participants were given a GitHub repo with the application files and task instructions. If participants got stuck during the session, hints specific to each tool interface were provided in links below the task instructions. Participants were told to think aloud during the session and given a prompt to continue thinking aloud if they fell silent for more than 15 seconds.

4   Evaluation Methodology

The video recordings were analyzed to manually count how frequently participants used the keyboard to enter a command (k), how frequently participants clicked (c), and how frequently participants moved their mouse and/or scrolled to click (m). Participant input type frequencies were stored in tuples as (k, c, m). Rather than count characters, we chose to count the number of total commands entered to normalize the input frequencies as we are neither counting the duration of the clicks nor the length of the mouse movements. In the OpenShift console, we counted commands as the number of times participants typed something followed by an enter or click to enter. Mouse movements were only noted when they were followed by a purposeful click (for example, moving the mouse to click on the browser back button), so random mouse movements were not counted in our analysis.

Specific input combinations were also noted such as how frequently participants copied and pasted text. Table X provides a list of all the input measures we collected for analysis and their exact definitions. Regular expressions were also used to create shortened descriptions of the patterns of input types used to complete the task with each tool. Table X provides a list of the regular expressions participants used to complete the task with each tool.

5   Results

P1	Copy/Paste	Keyboard	Click	Page Navigation
kubectl CLI	6	15	8	22
OpenShift console	0	2	77	80

P2	Copy/Paste	Keyboard	Click	Page Navigation
kubectl CLI	7	28	20	49
OpenShift console	0	13	145	172

P3	Copy/Paste	Keyboard	Click	Page Navigation
kubectl CLI	1	17	1	3
OpenShift console	4	5	142	192

P4	Copy/Paste	Keyboard	Click	Page Navigation
kubectl CLI	0	32	2	1
OpenShift console	4	14	133	137

P1	kubectl CLI	OpenShift console

P2	kubectl CLI	OpenShift console

P3	kubectl CLI	OpenShift console

P4	kubectl CLI	OpenShift console

6 Discussion

7 Future Work

8 Conclusion

References

[1] DILLON, E., ANDERSON, M., AND BROWN, M. Comparing mental models of novice programmers when using visual and command line environments. In *Proceedings of the 50th Annual Southeast Regional Conference* (2012), pp. 142–147.

[2] ERICSSON, K. A., AND SIMON, H. A. *Protocol analysis: Verbal reports as data.* the MIT Press, 1984.

[3] FEILD, H. A., ALLAN, J., AND JONES, R. Predicting searcher frustration. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2010), pp. 34–41.

[4] FOX, M. C., ERICSSON, K. A., AND BEST, R. Do procedures for verbal reporting of thinking have to be reactive? a meta-analysis and recommendations for best reporting methods. *Psychological bulletin 137*, 2 (2011), 316.

[5] GANDHI, I., AND GANDHI, A. Lightening the cognitive load of shell programming.

[6] GRAFSGAARD, J. F., WIGGINS, J. B., VAIL, A. K., BOYER, K. E., WIEBE, E. N., AND LESTER, J. C. The additive value of multimodal features for predicting engagement, frustration, and learning during tutoring. In *Proceedings of the 16th International Conference on Multimodal Interaction* (2014), pp. 42–49.

[7] HART, S. G., AND STAVELAND, L. E. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, vol. 52. Elsevier, 1988, pp. 139–183.

[8] MCDONALD, S., COCKTON, G., AND IRONS, A. The impact of thinking-aloud on usability inspection. *Proceedings of the ACM on Human-Computer Interaction 4*, EICS (2020), 1–22.

[9] TREHARNE, K., POWERS, D. M., AND LEIBBRANDT, R. Optimising visual and textual in search user interfaces. In *Proceedings of the 24th Australian Computer-Human Interaction Conference* (2012), pp. 616–619.

[10] UNAL, A., AND TOPU, F. B. Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high

school students. *Education and Information Technologies*, 1–19, year=2021, publisher=Springer.