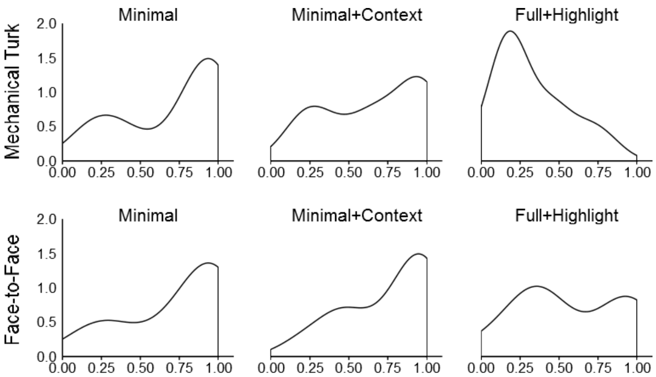

Figure 8. Density plots of the percentage of trials in which participants selected a "correct" anchor, for each Customization Layer variant, on MTurk (top) and in the lab (bottom). The minimal variants all have a higher right mode ("anchor search"), while F+H has higher left mode ("full panel search").

tion of the "valley" (local minimum) of the two distributions is also similar: 61% on MTurk, 63% in the lab.

This analysis points to a likely hidden variable in our data: the search strategy used by participants. Indeed the three Customization Layer conditions vary along this extra dimension, as shown in Figure 8. While these variations might explain the difference between the two minimal variants and F+H, the distributions for M and M+C are too similar to explain the difference in results observed between the two experiments.

For the next step in our exploratory analysis, we looked only at the data from participants who adopted an "anchor search" strategy to see if we could see any differences between M and M+C. More specifically, we re-ran the ANOVA on trial duration only on the data that fell above the middle point. For the MTurk experiment, M and M+C were both faster than Control ($p < .001$ each), and not significantly different from each other. For the lab experiment, M and M+C were also both faster than both F+H and Control (all $p < .05$). This secondary analysis suggests that the results of MTurk and the face-to-face experiments may be consistent, as long as you take search strategy into account: M and M+C are *both* faster than Control, and not significantly different from each other. As with any non-planned analysis, these results need to be interpreted with caution.

DISCUSSION

The results of our experiments are promising: whether on MTurk or in the lab, one of the minimal variants was significantly faster than Control, with a medium effect size, and no variant was slower. These performance improvements were obtained even though participants were exposed to Anchored Customization for the first time, and received little information to help them build an appropriate mental model. We now discuss the insights gathered on the three Customization Layer variants, and reflect on the possibilities offered by this new customization mechanism.

Reconciling the performance results. M and M+C performed differently in the two experiments. Our secondary analyses revealed a likely hidden variable, namely search strategy. For participants using the anchor search strategy, the results of the two experiments are consistent: M and M+C are *both* faster than Control, and not significantly different from each other. Hence there must be a significant difference in how the "full panel search" participants performed in the two experiments which translated into different relative performances for M and M+C on MTurk and in the lab. It might just be a statistical fluke. For instance, looking at the data shows that on MTurk all 5 of of the M+C participants that did a "full panel search" were disproportionately slower than the "full panel search" participants in *all* other conditions. This inconsistency suggests that the randomization of participant assignment to conditions may not have equalized individual differences. More research will be needed to further assess this issue.

Few participants in the F+H condition used the "anchor search" strategy: 2/12 on MTurk, 5/12 in the lab. It could be that highlighting settings in a panel is not a strong enough cue to help users acquire the Anchored Customization mental model. Since the structure of the settings panels is retained in F+H, there may be a strong transfer effect that encourages users to default to the "full panel search", instead of exploring anchors. Since few participants used F+H the way it was intended, we cannot conclude with certainty on its potential performance: is F+H necessarily slower than M and M+C, or can it be as fast when properly used? In any case, F+H is not slower than Control, so it would not be detrimental. Since some participants perceived the highlighting as a hint, F+H could be used as a "softer" version of Customization Layer for users who might not be comfortable with the degree of minimalism of the two minimal variants.

Performance/Awareness tradeoff. On MTurk, M was significantly faster than F+H and Control, but it also scored significantly lower on recognition of tab names. This could be interpreted as a performance/awareness tradeoff found in other multi-layered interfaces [13]. Yet, users in M did just as well as others in terms of recognizing the settings themselves. Thus, the awareness tradeoff here seems to affect only awareness of the *structure* of the upper layer (the full settings panel), not its content, as there were no differences in recognition scores for the settings themselves. This is not entirely surprising: in Customization Layer, all the settings are accessible via the minimal panel, albeit from different anchors. By contrast, the personalized interfaces studied by Findlater *et al.* [13] only display a static subset of features in the first layer, while the others were only visible in a different layer.

Applicability to other software. We focused our work on task management applications, and the question remains whether Anchored Customization could provide similar benefits for other types of software. At one extreme is text editors, which are often highly customizable. These editors typically have very few always-visible UI elements, and rely mostly on menus and keyboard shortcuts. Thus, they are not well-suited for Anchored Customization. Furthermore, text editor users are generally comfortable customizing their software by editing the config files directly. The opposite extreme is complex software applications designed for non-technical users, such as Adobe Creative Suite. These applications have many widgets that provide access to lots of features. Anchoring settings to these various UI elements is possible, but the resulting Customization Layer may be overwhelming. The main interface of these applications can itself be overwhelming, however, especially for new users. Anchored Customization would simply reflect the complexity of the underlying software.

Applicability to handheld devices. Our systematic review showed that mobile apps organize their settings in very similar ways to desktop applications. Mobile apps generally rely on icons and buttons for user input, since keyboard shortcuts and extensive menus are not available. As such, they are well suited for Anchored Customization. The limited screen real estate would warrant using a minimal variant. The anchored customization mechanism could be activated via a standard application menu, but touchscreens offer other possibilities: for instance, users could long-press or multi-tap an anchor to see its associated settings, or use a special standardized gesture to activate the Customization Layer. We observed that mobile applications generally offer fewer settings than desktop apps. The introduction of a well-designed customization mechanism could lead to more customizable mobile apps.

The developers' point of view. Beyond its benefits to users, our Anchored Customization approach may also change the way designers and developers think about customization. Although they are not *required* to change any setting to adopt Anchored Customization, the process of mapping settings to UI elements could have a positive effect on the settings offered. For instance, designers might realize that some parts of the interface have no setting anchored to them, which would highlight a potential opportunity for providing settings that cater to this area. Creating and labeling settings may also become faster, since the problem of finding appropriate words to refer to interface elements is mitigated by the context provided by the anchors. Finding categories to organize settings into tabs might also become superfluous.

Limitations

While our results are promising, our experiments had limitations, which come mostly from the challenges of evaluating customization in an artificial setting. We point out three. (1) To maximize ecological validity, we used Wunderlist's actual settings panel as a baseline for the Control condition. But some participants focused their feedback on its high quality visual design relative to our prototyped Customization Layer variants. In retrospect, we should have recreated this panel in the same visual style as our prototypes which might have increased the internal validity of our experiments. (2) The practice trial could have been more effective at conveying the intended model. Participants performed only one trial, thus only had to click on one anchor to complete it. However, to really understand the concept of Anchored Customization, one must click on at least two anchors to see that the settings offered are different. Some participants took time during the practice trial to explore the Customization Layer on their own, clicking on multiple anchors to see the outcome. This free exploration seemed more effective than our practice trial, and would also be more similar to real-world conditions. (3) Our experiments only included one application with a medium number of settings. It is possible that the number of settings offered by an app affects the relative performance of Customization Layer and settings panels.

We compared customization mechanisms mainly on how quickly participants could find a designated setting. However, in a real world situation, the *awareness* (or lack thereof) of which settings are available likely plays an important role. In traditional settings panels, awareness is gained by serendipitous discovery (also referred to as "incidental learning" [13]) : users happen to notice a setting of interest while searching the panel for another one. Serendipity can also happen in Anchored Customization, but the notion of proximity is relative to the anchors, instead of the settings panel's structure. Because our experiment task was time-constrained, these different forms of serendipity were not well captured.

CONCLUSIONS AND FUTURE WORK

Anchored Customization is an approach that places settings *in context* within the application interface, so that users are not required to learn the abstract structure of a settings panel. A systematic review of a set of Personal Task Management apps found that approximately 84% of the settings could be anchored in the UI. Our Customization Layer prototype reveals all the anchors as affordances for customization. We designed three variants of Customization Layer based on multi-layered interfaces, and implemented these variants on top of a popular web application for task management, Wunderlist. Two experiments (Mechanical Turk and face-to-face) showed that the two minimalist variants were 35-36% faster than Wunderlist's settings panel.

Evaluating the long term impact of this customization approach remains future work. A longitudinal field study would determine if a more usable customization mechanism does actually increases users' likelihood to customize. It would also help to verify our assumption that Anchored Customization requires one-time learning: once this approach is understood in the context of one particular app, the mental model should be transferable to other apps. Our prototype could be distributed to real users of Wunderlist as a browser extension, or adapted to other applications to compare the effect of different types of settings and different application domains.

Currently, app designers need to provide the mapping between settings and UI elements. With the growing popularity of advanced front-end frameworks in web development, code analysis techniques could possibly generate part of the mapping automatically, by determining which UI elements and listener functions are affected by a setting. Another possibility would be to involve users in the mapping process. Contrary to crowdsourced contextual help [7], users cannot be expected to *generate* the entire mapping themselves, but they could *tweak* a designer's mapping to better match their expectations. The idea of refining the mapping by aggregating data from individual users could be expanded to other customization opportunities as well. For instance, the most popular extensions and plugins could be anchored to the UI.

ACKNOWLEDGEMENT

This work was was funded by the Graphics Animation and New Media (GRAND) NCE and an NSERC Discovery grant.

REFERENCES

1. Roger Bakeman. 2005. Recommended effect size statistics for repeated measures designs. *Behavior Research Methods* 37, 3 (2005), 379–384. DOI: http://dx.doi.org/10.3758/BF03192707

2. Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. 2006. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 169–178. DOI: http://dx.doi.org/10.1145/1166253.1166280

3. Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, Noa Gradovitch, and Steve Whittaker. 2008. Improved Search Engines and Navigation Preference in Personal Information Management. *ACM Transactions on Information Systems* 26, 4, Article 20 (Oct. 2008), 24 pages. DOI: http://dx.doi.org/10.1145/1402256.1402259

4. Anastasia Bezerianos, Pierre Dragicevic, and Ravin Balakrishnan. 2006. Mnemonic Rendering: An Image-based Approach for Exposing Hidden Changes in Dynamic Displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 159–168. DOI: http://dx.doi.org/10.1145/1166253.1166279

5. Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. DOI: http://dx.doi.org/10.1191/1478088706qp063oa

6. Andrea Bunt, Cristina Conati, and Joanna McGrenere. 2007. Supporting Interface Customization Using a Mixed-initiative Approach. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI '07)*. ACM, New York, NY, USA, 92–101. DOI: http://dx.doi.org/10.1145/1216295.1216317

7. Parmit K. Chilana, Andrew J. Ko, and Jacob O. Wobbrock. 2012. LemonAid: Selection-based Crowdsourced Contextual Help for Web Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1549–1558. DOI: http://dx.doi.org/10.1145/2207676.2208620

8. Matjaz Debevc, Beth Meyer, Dali Donlagic, and Rajko Svecko. 1996. Design and evaluation of an adaptive icon toolbar. *User Modeling and User-Adapted Interaction* 6, 1 (1996), 1–21. DOI: http://dx.doi.org/10.1007/BF00126652

9. Morgan Dixon and James Fogarty. 2010. Prefab: Implementing Advanced Behaviors Using Pixel-based Reverse Engineering of Interface Structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1525–1534. DOI: http://dx.doi.org/10.1145/1753326.1753554

10. James R. Eagan, Michel Beaudouin-Lafon, and Wendy E. Mackay. 2011. Cracking the Cocoa Nut: User Interface Programming at Runtime. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 225–234. DOI: http://dx.doi.org/10.1145/2047196.2047226

11. W. Keith Edwards, Scott E. Hudson, Joshua Marinacci, Roy Rodenstein, Thomas Rodriguez, and Ian Smith. 1997. Systematic Output Modification in a 2D User Interface Toolkit. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST '97)*. ACM, New York, NY, USA, 151–158. DOI: http://dx.doi.org/10.1145/263407.263537

12. Leah Findlater and Joanna McGrenere. 2008. Impact of Screen Size on Performance, Awareness, and User Satisfaction with Adaptive Graphical User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1247–1256. DOI: http://dx.doi.org/10.1145/1357054.1357249

13. Leah Findlater and Joanna McGrenere. 2010. Beyond performance: Feature awareness in personalized interfaces. *International Journal of Human-Computer Studies* 68, 3 (2010), 121 – 137. DOI: http://dx.doi.org/10.1016/j.ijhcs.2009.10.002

14. Gerhard Fischer. 1993. *Adaptive User Interfaces*. Elsevier Science Publishers B.V., Chapter Shared Knowledge in Cooperative Problem-Solving Systems – Integrating Adaptive and Adaptable Components, 49–68. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.310.8392

15. Gerhard Fischer and Elisa Giaccardi. 2006. Meta-design: A Framework for the Future of End-User Development. In *End User Development*, Henry Lieberman, Fabio Paternò, and Volker Wulf (Eds.). Human-Computer Interaction Series, Vol. 9. Springer Netherlands, 427–457. DOI: http://dx.doi.org/10.1007/1-4020-5386-X_19

16. Stephen Fitchett, Andy Cockburn, and Carl Gutwin. 2013. Improving Navigation-based File Retrieval. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2329–2338. DOI: http://dx.doi.org/10.1145/2470654.2481323

17. G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The Vocabulary Problem in Human-system Communication. *Commun. ACM* 30, 11 (Nov. 1987), 964–971. DOI: http://dx.doi.org/10.1145/32206.32212

18. Alfonso García Frey, Gaëlle Calvary, and Sophie Dupuy-Chessa. 2010. Xplain: An Editor for Building Self-explanatory User Interfaces by Model-driven Engineering. In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '10)*. ACM, New York, NY, USA, 41–46. DOI: http://dx.doi.org/10.1145/1822018.1822026
19. 6 Wunderkinder GmbH. 2015. Wunderlist. Version 3.12.3. (2015). http://www.wunderlist.com (Retrieved on 2015-07-01).
20. Mona Haraty, Diane Tam, Shathel Haddad, Joanna McGrenere, and Charlotte Tang. 2012. Individual Differences in Personal Task Management: A Field Study in an Academic Setting. In *Proceedings of Graphics Interface 2012 (GI '12)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 35–44. http://dl.acm.org/citation.cfm?id=2305276.2305284
21. Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. 1998. The LumièRe Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 256–265. http://dl.acm.org/citation.cfm?id=2074094.2074124
22. J. Johnson, T.L. Roberts, W. Verplank, D.C. Smith, C.H. Irby, M. Beard, and K. Mackey. 1989. The Xerox Star: a retrospective. *Computer* 22, 9 (Sept 1989), 11–26. DOI: http://dx.doi.org/10.1109/2.35211
23. Helge Kahler, Anders Mørch, Oliver Stiemerling, and Volker Wulf. 2000. Tailorable systems and cooperative work. *Computer Supported Cooperative Work (CSCW)* 9, 1 (2000), 1–4. DOI: http://dx.doi.org/10.1023/A:1017243824820
24. Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. 2006. End-User Development: An Emerging Paradigm. In *End User Development*, Henry Lieberman, Fabio Paternò, and Volker Wulf (Eds.). Human-Computer Interaction Series, Vol. 9. Springer Netherlands, 1–8. DOI: http://dx.doi.org/10.1007/1-4020-5386-X_1
25. Wendy E. Mackay. 1991. Triggers and Barriers to Customizing Software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. ACM, New York, NY, USA, 153–160. DOI: http://dx.doi.org/10.1145/108844.108867
26. Thomas W. Malone, Kum-Yew Lai, and Christopher Fry. 1995. Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. *ACM Transactions on Information Systems* 13, 2 (April 1995), 177–205. DOI: http://dx.doi.org/10.1145/201040.201047
27. Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Patina: Dynamic Heatmaps for Visualizing Application Usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 3227–3236. DOI: http://dx.doi.org/10.1145/2470654.2466442
28. Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. 2002. An Evaluation of a Multiple Interface Design Solution for Bloated Software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 164–170. DOI: http://dx.doi.org/10.1145/503376.503406
29. Xiaojun Meng, Shengdong Zhao, Yongfeng Huang, Zhongyuan Zhang, James Eagan, and Ramanathan Subramanian. 2014. WADE: Simplified GUI Add-on Development for Third-party Software. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2221–2230. DOI: http://dx.doi.org/10.1145/2556288.2557349
30. Ben Shneiderman. 2003. Promoting Universal Usability with Multi-layer Interface Design. In *Proceedings of the 2003 Conference on Universal Usability (CUU '03)*. ACM, New York, NY, USA, 1–8. DOI: http://dx.doi.org/10.1145/957205.957206
31. Daniel J Simons and Christopher F Chabris. 1999. Gorillas in our midst: Sustained inattentional blindness for dynamic events. *Perception-London* 28, 9 (1999), 1059–1074.
32. Gunnar Stevens and Torben Wiedenhöfer. 2006. CHIC - a Pluggable Solution for Community Help in Context. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles (NordiCHI '06)*. ACM, New York, NY, USA, 212–221. DOI: http://dx.doi.org/10.1145/1182475.1182498
33. Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. 2006. User Interface Façades: Towards Fully Adaptable User Interfaces. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 309–318. DOI: http://dx.doi.org/10.1145/1166253.1166301
34. Gaurav Paruthi Tao Dong, Mark S. Ackerman, Mark W. Newman. 2013. Social Overlays: Collectively Making Websites More Usable. Lecture Notes in Computer Science, Vol. 8120. Springer Berlin Heidelberg, Berlin, Heidelberg. DOI: http://dx.doi.org/10.1007/978-3-642-40498-6
35. Volker Wulf and Björn Golombek. 2001. Direct activation: A concept to encourage tailoring activities. *Behaviour & Information Technology* 20, 4 (2001), 249–263. DOI: http://dx.doi.org/10.1080/01449290110048016