

Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces

Leah Findlater and Krzysztof Z. Gajos

■ *Adaptive graphical user interfaces (GUIs) have the potential to improve performance and user satisfaction by automatically tailoring the presentation of functionality to each individual user. In practice, however, many challenges exist, and evaluation results of adaptive GUIs have been mixed. To guide researchers and designers in developing effective adaptive GUIs, we outline a design space and discuss three important aspects to consider when conducting user evaluations of these types of interfaces: the control and reporting of adaptive algorithm characteristics, the impact of task choice and user characteristics on the overall effectiveness of a design, and evaluation measures that are appropriate for adaptive interaction.*

Adaptive graphical user interfaces (GUIs) automatically tailor the presentation of functionality to better fit an individual user's tasks, usage patterns, and abilities. A familiar example of an adaptive interface is the Windows XP start menu, where a small set of applications from the "All Programs" submenu is replicated in the top level of the "Start" menu for easier access, saving users from navigating through multiple levels of the menu hierarchy (figure 1). The potential of adaptive interfaces to reduce visual search time, cognitive load, and motor movement is appealing, and when the adaptation is successful an adaptive interface can be faster and preferred in comparison to a nonadaptive counterpart (for example, Gajos et al. [2006], Greenberg and Witten [1985]).

In practice, however, many challenges exist, and, thus far, evaluation results of adaptive interfaces have been mixed. For an adaptive interface to be successful, the benefits of correct adaptations must outweigh the costs, or usability side effects, of incorrect adaptations. Often, an adaptive mechanism designed to improve one aspect of the interaction, typically motor movement or visual search, inadvertently increases effort along another dimension, such as cognitive or perceptual load. The result is that many adaptive designs that were expected to confer a benefit along one of these dimensions have failed in practice. For example, a menu that tracks how frequently each item is used and adaptively reorders itself so that items appear in order from most to least frequently accessed should improve motor performance, but in reality this design can slow users down and reduce satisfaction because of the constantly changing layout (Mitchell and Schneiderman [1989]; for example, figure 2b). Commonly cited issues with adaptive interfaces include the lack of control the user has over the adaptive process and the difficulty that users may have in predicting what the system's response will be to a user action (Höök 2000).

User evaluation of adaptive GUIs is more complex than eval-

uation of traditional, nonadaptive interfaces due to the increased variability of interaction. To guide researchers and designers in developing effective adaptive GUIs, particularly adaptive control structures such as menus and toolbars, we first briefly summarize the design space of these interfaces before discussing three evaluation issues in detail: (1) the control of adaptive algorithm characteristics and their potential impact; (2) the user's familiarity with the task and interface; (3) measures that are appropriate for adaptive interaction, particularly the impact of adaptive behavior on overall awareness of features in the interface. The focus of this article is on GUI control structures, such as menus and toolbars, and we specifically do not cover systems that adapt information content to the user's needs, such as recommender systems.

Relationships to the Theme Articles

This article is related to all three theme articles in this issue. In terms of the usability benefits of intelligence (Lieberman, in this issue), graphical adaptive user interfaces have the promise of allowing technology to adapt itself to the users, making efficient use of user input, and providing methods of input and output that are natural for the user. We also include many examples of the general concepts and analysis patterns introduced in the usability side-effects theme article (Jameson, in this issue), but by focusing on a specific type of system intelligence, we are able to offer a more tightly knit analysis, including some well-established empirical results and concepts that are specific to graphical adaptive user interfaces but that may offer general insights for the design of other types of intelligent interactive systems. Finally, our analysis illustrates some of the subtleties involved in the evaluation of interactive intelligent systems (see the theme article on usability engineering methods) that result from trade-offs among usability goals and from the influences of properties of users, tasks, and contexts.

Design Space of Adaptive GUIs

The presentation of menu and toolbar items in the user interface can be adapted to a particular person through several automated means. For example, figure 2 shows some examples of adaptation of a pull-down menu where the adaptive algorithm has predicted that items F and D are the most likely to be needed next by the user. One presentation option is to change the spatial arrangement of the menu and toolbar items to make them easier to access. This can be done by moving (figure 2b and e), resizing (figure 2c), or replicating (figure 2f) adaptively predicted items, or by hiding those items that are predicted to be the least useful (fig-

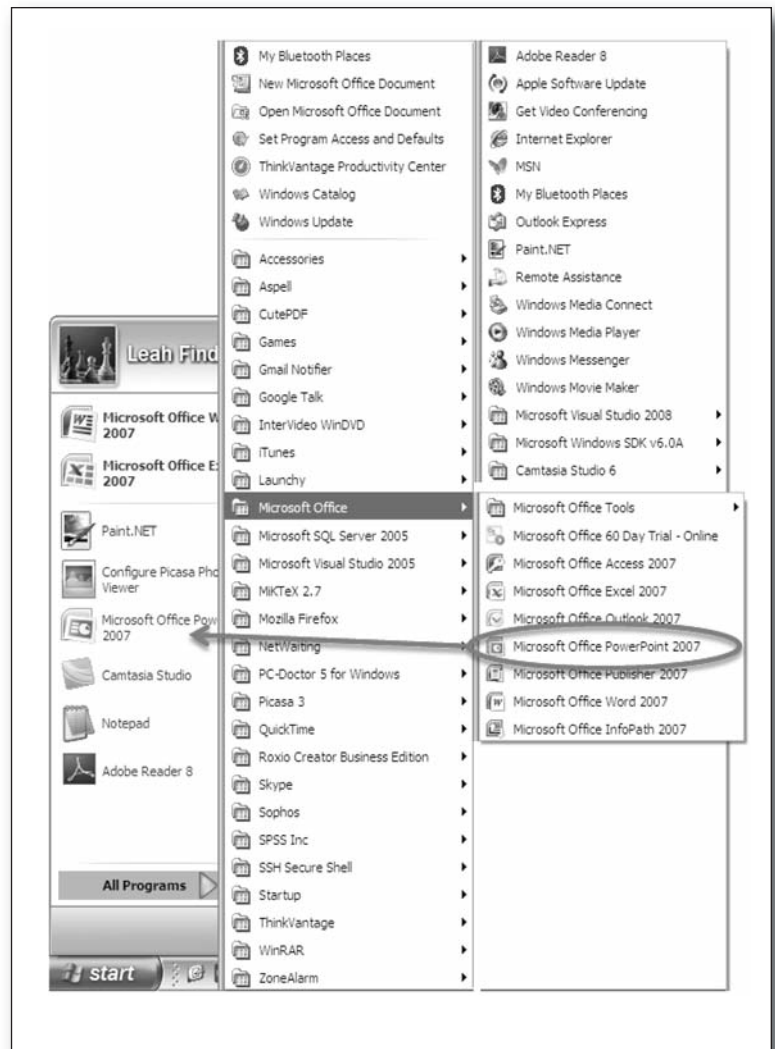


Figure 1. Windows XP Start Menu.

In this example, Microsoft PowerPoint has been adaptively copied to the top level of the menu for easier access.

ure 2d). Another option is to alter the visual appearance of items to make some easier to find. Color highlighting, for example, has been proposed to draw the user's attention to adaptively predicted items (figure 2g). Finally, one can change the dynamic behavior of the interface to draw visual attention to some elements. For example, with ephemeral adaptation (figure 2h), those items predicted to be the most likely to be needed by the user appear abruptly to draw the user's attention when the menu opens, while the remaining items gradually fade in.

Each type of adaptation has its own set of strengths and weaknesses. Spatial adaptation can be particularly effective when it greatly reduces the amount of navigation required to reach items (for example, in a hierarchical menu structure: Greenberg and Witten [1985]), or when adapting the

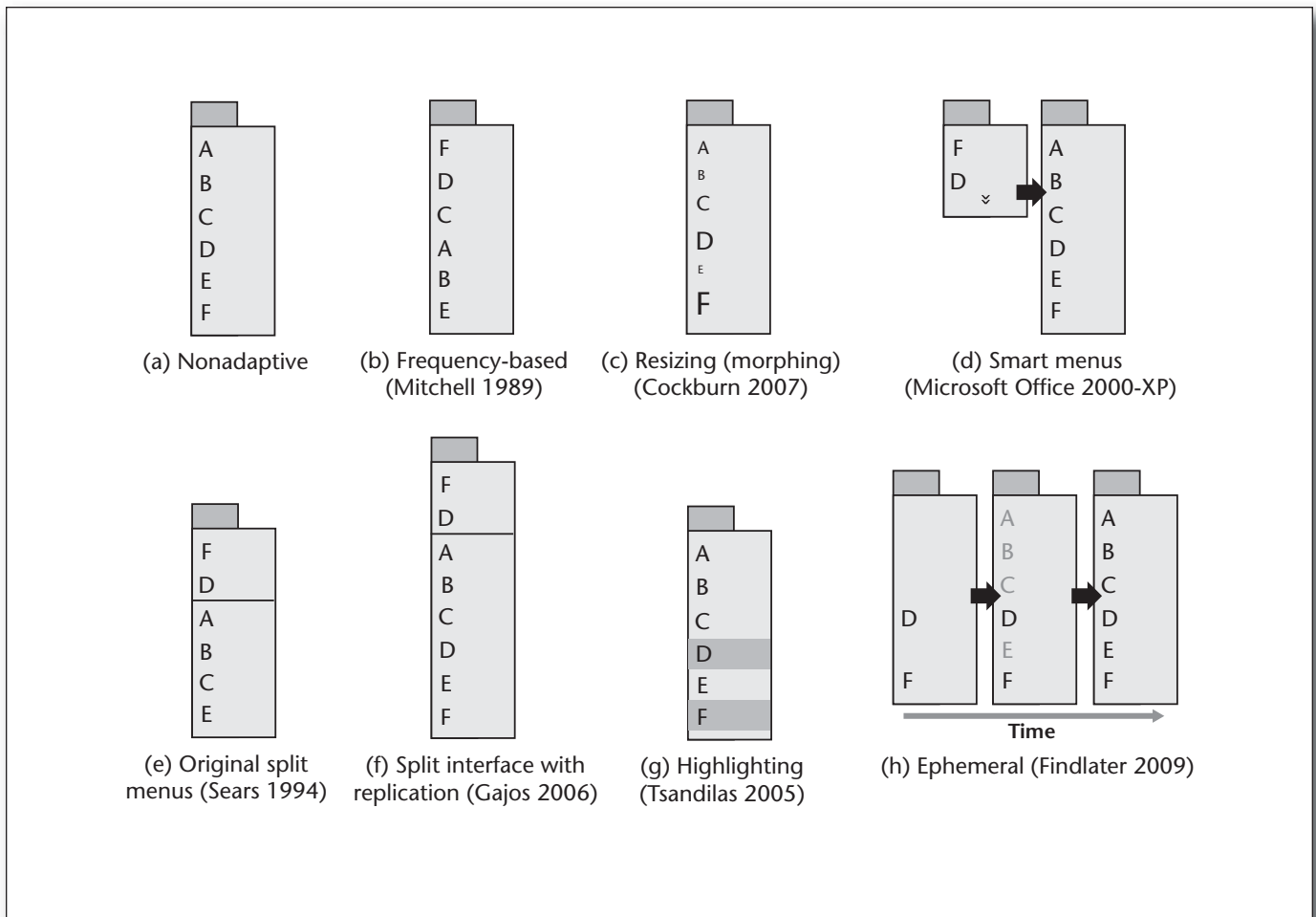


Figure 2. Examples of Adaptive Interface Strategies Applied to Pull-Down Menus.

interface to support the motor abilities of individual users (Gajos, Wobbrock, and Weld 2008). The downside, however, is that it introduces spatial instability into the interface, so items may not be located where the user expects them to be. Besides increasing potential for frustration, some types of spatial adaptation may prevent users from improving the efficiency of their visual search thus directly hurting performance (Cockburn, Gutwin, and Greenberg 2007).

In contrast, adapting the visual appearance of items only has the potential to reduce the visual search component of item selection, but the advantage is that it maintains spatial stability. Of techniques that adapt visual appearance of items, color highlighting (figure 2g) has received the widest amount of research attention, and, unfortunately, evaluations have not shown it to provide a performance benefit over a nonadaptive interface (Gajos et al. 2006; Findlater 2009). In comparison, the dynamic behavior of ephemeral adaptation (figure 2h) also reduces visual search time and has been shown to improve performance (Findlater 2009).

One of the more important design decisions is whether the use of adaptation should be elective or mandatory, that is, whether the user is able to ignore the adaptation or not. The split menu shown in figures 2e and 2f provide an example of elective versus mandatory adaptation. In the original design of figure 2e (Sears 1994), the few (up to 4) most frequently used elements were moved to the top of the menu. If this adaptation occurs frequently, it may prevent users from developing spatial memory of the menu layout and can consequently impair performance and user satisfaction in comparison to a nonadaptive menu (Findlater and McGrenere 2004). An alternative is to allow the use of adaptive predictions to be elective: in figure 2f, where adaptively predicted items are copied rather than moved to the top of the menu, the user can choose either to scan the adaptive predictions or simply to skip to known locations in the main interface below. The elective split interface design has been shown to be faster and preferred in comparison to the original mandatory design (Gajos et al. 2006). One of the potential drawbacks of elec-

tive approaches, however, is that users have to put extra cognitive effort into discovering whether a useful adaptation has taken place before deciding whether to take advantage of it. For example, in the Windows XP Start Menu (figure 1), an explicit glance is necessary to see if the desired program has been copied to the top-level menu or whether the “all programs” submenu has to be opened.

Therefore, a related factor to consider is the locality of the adaptation, or where the adaptation appears relative to the user’s current focus. When the adaptation occurs close to or at the user’s point of focus, it makes it easier for people to serendipitously discover a helpful adaptation. Adaptation approaches that change the visual appearance of items, such as color highlighting (figure 2g), should result in high (close) locality because they occur at the location where the user would look by default. In comparison, with low locality, where the adaptation occurs farther from the user’s point of focus, the user must make a more conscious effort to take advantage of the adaptation. Locality is especially important to consider with spatial adaptation. As a case in point, when the replicated split interface approach of figure 2f was applied to toolbar adaptations, people ignored helpful adaptations between 7 percent and 19 percent of the time (Gajos et al. 2006), which may have been partly because the adaptively predicted items appeared far from the location of those items in the main interface.

Adaptive Algorithm Characteristics

A common pitfall in evaluating adaptive interfaces is the lack of experimental control over adaptive algorithm characteristics. Here, we discuss three characteristics that researchers have begun to isolate: the accuracy of the adaptive algorithm’s predictions, the predictability of the algorithm’s behavior, and the frequency with which adaptations are introduced in the interface. These characteristics should be incorporated where possible into study designs, and our experience in designing and evaluating adaptive

GUIs emphasizes the importance of considering all three in combination.

Adaptive accuracy is the accuracy with which the adaptive algorithm predicts which item the user will need next. Adaptive GUIs that can predict the user’s needs with higher accuracy can make users faster and can result in greater utilization of adaptive predictions by the user (Findlater and McGrenere 2008; Gajos et al. 2006; Gajos et al. 2008; Tsandilas and Schraefel 2005). However, the effectiveness of individual adaptive designs may also interact with accuracy: Tsandilas and Schraefel (2005) found that lower adaptive accuracy affected the user’s error rate, but that this was true for only one of two adaptive menu designs they studied.

Predictability and frequency of change have not been as well studied as accuracy. Both Cockburn, Gutwin, and Greenberg (2007) and Bridle and McCreath (2006) have highlighted the importance of considering frequency of change in evaluations of adaptive interfaces (Bridle and McCreath called it “predictability” and Cockburn and colleagues called it “stability”). As a comparative example, the layout of the original split menus (Sears 1994) only changed once during the course of the study and resulted in a performance benefit over a traditional, static menu; however, follow-up work (Findlater and McGrenere 2004) showed that more frequently changing the layout of the split menu (as often as after every selection by the user) decreased performance.

In comparison to frequency of change, predictability may have a more subjective component. Gajos et al. (2008) define predictability as the ease with which users can understand and predict the behavior of the adaptive algorithm. In general, an interface that adapts less frequently should be more predictable for the user; however, this is not necessarily the case and it may be useful to distinguish between the two concepts. For example, adaptive changes based strictly on the set of most recently used features may be easily understood by the user, making the adaptive behavior predictable (as shown in Gajos et al. 2008), but depending on the specific

sequence of features used, this approach may not necessarily offer low frequency of change.

Isolating either accuracy, predictability, or frequency of change as a factor in a study design can be challenging. For example, Tsandilas and Schraefel (2005) compared the effect of three levels of accuracy for adaptively predicted items in a list. However, their experimental system highlighted a randomly chosen subset of items in trials that were incorrectly predicted, so the lower accuracy conditions (that is, with more incorrect predictions) may have appeared more unpredictable than the higher accuracy conditions. In a study of adaptive toolbars, Gajos et al. (2006) used two different experimental tasks (selection distributions) that resulted in different levels of accuracy for the same underlying adaptive algorithm. Again, this did not purely isolate accuracy, because the tasks may have affected other aspects of adaptive behavior. One approach to managing this issue is to use an identical set of selections for each condition and determine adaptive predictions using two steps: (1) applying a base algorithm, and (2) adjusting predictions to achieve a desired level of accuracy while adhering to several constraints (for example, to maintain the level of predictability where possible) (Findlater and McGrenere 2008). Even with careful study design, however, it may not always be possible to isolate the impact of specific adaptive algorithm characteristics on the user’s performance and satisfaction. In such cases, stating the characteristics of the adaptive algorithm when reporting on evaluations will at least allow readers to interpret the results more easily.

Tasks and User Expertise

As is discussed in the usability side-effects theme article, the best way of realizing intelligence in an interactive system may depend strongly on properties of the individual users. Adaptive user interfaces offer many examples.

The user’s familiarity with the task and application can affect the overall effectiveness of an adaptive interface. Some adaptive mechanisms may be

helpful to novices but not expert users (Trevelyan and Browne 1987) and may even hinder performance for experts. When conducting evaluations, it may be necessary in some cases to let users develop expertise with the interface and to see whether this affects the usefulness of the adaptive mechanism. Personality traits related to control (Goren-Bar et al. 2005) and whether users are feature-keen or feature-shy with respect to having extra, unused items in the interface (McGrenere, Baecker, and Booth 2002) may all affect the success of a system that provides adaptation.

Individual users' motor and cognitive abilities may also affect the overall benefit of an adaptive interface. Gajos, Wobbrock, and Weld's (2008) SUPPLE system adjusts the layout, size, and structure of interface elements to accommodate both device and user characteristics. An evaluation with 17 participants showed that adapting the interface based on automatic detection of motor abilities made motor-impaired users faster and more accurate than with a manufacturer's default interface layout, and it was strongly preferred (Gajos, Wobbrock, and Weld 2008). This approach adapts the interface only once at the outset, but it does highlight a cost/benefit trade-off based on the user's abilities: while both motor-impaired and able-bodied users were faster with their personalized interfaces, the able-bodied users, who did not achieve as large a performance gain, preferred the aesthetics of the original interface presentation.

The pace of interaction with the interface and the cognitive complexity of the task can also influence what aspects of the adaptive interface users find relevant. For example, Findlater and McGrenere (2008) found that adaptive split menus used on a small, PDA-sized screen provided greater performance and user satisfaction improvements in comparison to the same menus on a desktop-sized screen because of the relative difficulty of navigating the interface on the smaller screen. As another example, Gajos et al. (2006) conducted two experiments with adaptive GUIs: the first experiment used an open-ended and creative task, while the second experiment

used a tightly controlled task that required fast and largely mechanical selections of toolbar items. User comments during the first experiment suggested that users valued an adaptive approach that allowed related functionality to remain spatially grouped, allowing them to concentrate on the task. In contrast, the tightly controlled task used in the second experiment caused users to pay more attention to the operational properties of the interface, resulting in a more positive response to an adaptive approach that reduced the mechanical movement required to reach items.

Evaluation Measures

One of the goals of adaptive interfaces is to present the user interface functionality in such a way as to increase task efficiency, which can be measured using task completion times and error rates. User satisfaction measures are also important, and can include questions specific to adaptive interfaces, such as rating the obtrusiveness, the predictability, and the usefulness of the adaptive behavior.

One aspect of adaptive interaction that is not explicitly captured using these standard evaluation measures is the impact that working in an adaptive interface can have on the user's overall awareness of features in the interface, in other words, the impact on the user's breadth of experience. When an interface adapts to make useful features more accessible for a user's current task, there may be a negative impact on the user's awareness of the full set of available features, which, in turn, can make future tasks more difficult. To evaluate this aspect of adaptive interfaces, the user's level of awareness of the full set of features in the interface should be assessed alongside traditional speed and user satisfaction measures. Awareness can be measured through a recognition test of features in the interface (Findlater and McGrenere 2007).

Experimental results have shown a measurable trade-off between efficiency and the user's awareness of unused features in the interface. Studying adaptive split menus, Findlater and McGrenere (2008) found that adaptive

menus that predicted the user's needs with a high degree of accuracy were efficient to use, but because they better focused the user's attention on only a subset of features, they resulted in low awareness of unused features. In contrast, the adaptive split menus that were not as accurate at predicting the user's needs were not as efficient, but resulted in higher awareness. Reduced awareness of advanced features in the interface may reduce the user's longer-term learning and efficiency: a follow-up study by Findlater and McGrenere showed that high adaptive accuracy also negatively affects users' performance when they are asked to complete new tasks (Findlater et al. 2009).

What is considered to be a desirable balance between efficiency and awareness may change in different design contexts. High awareness of advanced features may be more important for software applications where users are expected to mature into experts, for example, as with a complex integrated development environment. An adaptive mechanism that could predict new, potentially useful features for users may be beneficial in these situations. On the other hand, for applications that are used on a less frequent basis or for those applications that cater to a range of users with varying levels of expertise, efficiency for core tasks may be more important than awareness.

Conclusion

Adaptive user interfaces have the potential to improve performance and user satisfaction, but evaluation results have been mixed. In our own research, we have found that two adaptive interface designs have provided more consistently positive results: elective split interfaces (Gajos et al. 2006) and ephemeral adaptation (Findlater 2009). Elective split interfaces (a generalization of split menus) constrain the adaptation such that useful interface elements are copied to a designated section of the interface (for example, top of the menu, a special toolbar, or a separate pane), and improve user performance in large or hierarchical user interfaces while allowing users to develop familiarity with the layout of

the nonadaptive section of the interface. Ephemeral adaptation uses abrupt onset of a few elements in a menu or a popup window to draw the user's attention to those elements while the remaining items fade in gradually yet quickly enough not to hinder performance.

We have outlined a design space of adaptive interfaces and discussed the unique challenges of evaluating adaptive interfaces. Three important aspects to consider when conducting evaluations include the control and reporting of adaptive algorithm characteristics, the impact of task choice and user characteristics on the overall effectiveness of a design, and evaluation measures that are appropriate for adaptive interaction. Although we have focused on purely adaptive approaches, the points made here are also relevant to mixed-initiative approaches to adapting the interface, where control over the adaptation is shared between the system and the user (see the case study by Bunt, Conati, and McGrenere in this issue).

References

- Bridle, R., and McCreath, E. 2006. Inducing Shortcuts on a Mobile Phone Interface. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, 327–329. New York: Association for Computing Machinery.
- Bunt, A.; Conati, C.; and McGrenere, J. A Mixed-Initiative Approach to Interface Personalization. *AI Magazine* 30(4).
- Cockburn, A.; Gutwin, C.; and Greenberg, S. 2007. A Predictive Model of Menu Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*, 627–636. New York: Association for Computing Machinery
- Findlater, L., and McGrenere, J. 2004. A Comparison of Static, Adaptive and Adaptable Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*, 89–96. New York: Association for Computing Machinery.
- Findlater, L., and McGrenere, J. 2007. Evaluating Reduced-Functionality Interfaces According to Feature Findability and Awareness. In *Proceedings of IFIP International Conference on Human-Computer Interaction*, 592–605. Berlin: Springer.
- Findlater, L., and McGrenere, J. 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)*, 1247–1256. New York: Association for Computing Machinery.
- Findlater, L., and McGrenere, J. 2009. Beyond Performance: Feature Awareness in Personalized Interfaces. *International Journal of Human Computer Studies*. In press.
- Findlater, L.; Moffatt, K.; McGrenere, J.; and Dawson, J., 2009. Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Deletion Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*, 1655–1664. New York: Association for Computing Machinery.
- Gajos, K. Z.; Czerwinski, M.; Tan, D. S.; and Weld, D. S. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'06)*, 201–208. New York: Association for Computing Machinery.
- Gajos, K. Z.; Everitt, K.; Tan, D.S.; Czerwinski, M.; and Weld, D.S. 2008. Predictability and Accuracy in Adaptive User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*, 1271–1274. New York: Association for Computing Machinery.
- Gajos, K. Z.; Wobbrock, J. O.; and Weld, D. S. 2008. Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*, 1257–1266. New York: Association for Computing Machinery.
- Goren-Bar, D.; Graziola, I.; Pianesi, F.; and Zancanaro, M. 2005. The Influence of Personality Factors on Visitor Attitudes Towards Adaptivity Dimensions for Mobile Museum Guides. *User Modeling and User Adapted Interaction* 16(1): 31–62.
- Greenberg, S., and Witten, I. 1985. Adaptive Personalized Interfaces: A Question of Viability. *Behaviour and Information Technology* 4(1): 31–45.
- Höök, K. 2000. Steps to Take before Intelligent User Interfaces Become Real. *Journal of Interacting with Computers* 12(4): 409–426.
- Jameson, A. Understanding and Dealing with Usability Side Effects of Intelligent Processing. *AI Magazine* 30(4).
- Lieberman, H. User Interface Goals, AI Opportunities. *AI Magazine* 30(4).
- McGrenere, J., and Moore, G. 2000. Are We All in the Same "Bloat"? In *Proceedings of Graphics Interface*, 187–196. Toronto: Canadian Human-Computer Communications Society.
- McGrenere, J.; Baecker, R.; and Booth, K. 2002. An Evaluation of a Multiple Interface Design Solution for Bloated Software. *CHI Letters* 4(1): 163–170.
- Mitchell, J., and Shneiderman, B. 1989. Dynamic Versus Static Menus: An Exploratory Comparison. *SIGCHI Bulletin* 20(4): 33–37.
- Sears, A., and Shneiderman, B. 1994. Split Menus: Effectively Using Selection Frequency to Organize Menus. *ACM Transactions on Computer-Human Interaction* 1(1): 27–51.
- Shneiderman, B., and Maes, P. 1997. Direct Manipulation Versus Interface Agents: Excerpts from Debates at IUI 97 and CHI 97. *Interactions* 4(6): 42–61.
- Spaulding, A. Usability Engineering for Interactive Intelligent Systems. *AI Magazine* 30(4).
- Trevelyan, R., and Browne, D. P. 1987. A Self-Regulating Adaptive System. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI'87)*, 103–107. New York: Association for Computing Machinery.
- Tsandilas, T., and Schraefel, M. C. 2005. An Empirical Assessment of Adaptation Techniques. In *CHI'05 Extended Abstracts*, 2009–2012. New York: Association for Computing Machinery.

Leah Findlater is a postdoctoral fellow in the Information School at the University of Washington. Her research interests span a range of topics in human-computer interaction, including personalization, accessibility, and information and communication technologies for development (ICTD). She received her Ph.D. in computer science from the University of British Columbia in 2009 and currently holds a Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship.

Krzysztof Z. Gajos is an assistant professor of computer science at the Harvard School of Engineering and Applied Sciences. Krzysztof is primarily interested in intelligent interactive systems, an area that spans human-computer interaction, artificial intelligence, and machine learning. Krzysztof received his B.Sc. and M.Eng. degrees in computer science from the Massachusetts Institute of Technology. Subsequently he worked as a research scientist at the MIT Artificial Intelligence Laboratory, where he managed the Intelligent Room Project. In 2008, he received his Ph.D. in computer science from the University of Washington in Seattle. Before coming to Harvard, he spent a year as a postdoctoral researcher with the Adaptive Systems and Interaction group at Microsoft Research.