Adaptive Graphical User Interfaces: Accomplishments, Challenges, and Future Directions

Sylvie Foss
fossyl@cs.ubc.ca

1. INTRODUCTION

With the ever increasing ubiquity of computers, tablets, smart phones, and other digital devices, graphical user interfaces are reaching a wider audience than ever before. As this user population is highly variable in capabilities, experience, age, and other characteristics, personalization is essential to ensure everybody can interact with their device. User needs can also be affected by environmental factors, or their personal goals at any given time. Since these conditions typically change over time, an adaptive approach is needed. Adaptation can also be beneficial when a user is learning how to use an interface, incrementally teaching them how to use the software. This prevents them having to spend significant amounts of time pouring over help materials and doing tutorial exercises, or even working through trial-and-error. Prior work in this field includes interfaces to support users with disabilities, the definition and study of evaluation metrics, exploration of various adaptation methods, and the implementation of adaptive tutoring systems. Each makes important contributions to the study of adaptive interfaces, and exposes challenges and potential for future work.

2. KEY CONSIDERATIONS

2.1 Abilities and Prior Knowledge

There is a wide variety of abilities across the user population, ranging from proficient computer users with perfect vision to casual technology users with limited motor control, and everything in between. A common disability affecting interaction with screens is reduced eyesight, particularly with age. User characteristics can also affect usability. Handedness when using a pen tablet determines which portion of the display will be obstructed by the user’s hand. Those with large fingers will have increased trouble accessing small touch icons. There is also the difference between experienced users of an application and those using it for the first time. Transferability of applicable background knowledge, if available, also factors into how easily a user can learn a new interface.

2.2 What and When to Adapt

Due to the range of possible challenges faced by users, adaptation ability of software can be key to user performance and satisfaction. Given that the software, not the user, must adapt, the key consideration becomes what to adapt and when [3]. If the software adapts to make the wrong tools more easily accessible, the user could accidentally select the wrong item and face the cost of undoing that mistake, or simply be confused by the increased saliency of inappropriate elements. If the correct items are added to the adaptive space at the wrong time, the user could be interrupted or irritated, wasting time handling the interruption or even dismissing the adaptive component altogether out of frustration.

3. ADAPTATION PARADIGMS

3.1 Adaptive Toolbars and Menus

A GUI with an adaptive toolbar or menu has an adaptive area in which frequently used items are placed. Placement can be by duplication of the item, leaving the original location unaffected and merely adding a copy of the item to the adaptive area. Alternatively the item can be moved to the adaptive area, removing it from its original location. For frequently used menu items, the adaptive area can be a visually separated segment of the same menu. Those items could also be placed in an always-visible adaptive toolbar, which has the advantage of making the frequently used items visible at all times without having to open a menu. Items from always-visible toolbars could also be duplicated or moved to another always-visible adaptive area, but this is rarely used as the visual duplication can be confusing for users. While these are some of the more popular options, design is limited only by the GUI developer.

3.2 Incremental Interfaces

Novice users of a system benefit from appropriate help at the correct times, simplifying the learning process required to get started with the software. This can come in the form of helpful hints teaching the user how to complete a task, an initially simple UI that gains in complexity over time and use, or other possibilities. The key challenges here are what to give the user and when to provide it.

3.3 Accessible Devices and Environment

Touch screens can pose an especially great challenge for disabled users versus a desktop computer. Motor control limitations can make interaction difficult, as precise gestures and small touch targets are involved. Furthermore, accidentally making the wrong gesture or hitting the wrong target can be very disruptive to the user’s workflow. Poor eyesight can make any screen difficult to view, but even a user with no vision impairments can still struggle in low light conditions. Devices can include sensor hardware to observe environmental factors such as lighting conditions, noise, movement in the case of a mobile user or commuter, and other factors affecting usability. Combined with the software tracking of user interactions, a system can adapt to multiple conditions and be accessible for users of varying physical abilities.

3.4 Interest Mining

Adaptive approaches have also been applied in entertainment contexts. Since these are not productivity tools, the interface merely serves as a way for the user to locate something that interests them. In contrast to GUIs of productivity software, the user is often more interested in discovering new items than locating known items. This being the case, it is more acceptable to swap out most of the GUI items for those the software has deemed interesting for the user based on data mining the user's history with the software. A prime example is the Netflix interface, in which the choice of movie categories is entirely based on viewing history.

4. PRIOR WORK

4.1 Microsoft Adaptive Menus

The split menu concept appeared in Windows XP in the form of an adaptive Start Menu. Shortcuts to frequently opened applications are added to the adaptive area of the menu so that users need not traverse multi-level menus in order to access their most used programs. This concept has persisted over the years and is still present today in Windows 7. Formal evaluation of this menu have not been carried out, but there is significant anecdotal evidence in favor of the design [5]. A similar pattern is seen in other Microsoft products, such as Office 2010 and PowerPoint 2010. In these cases, the adaptive menu is populated with documents the user has recently edited.

4.2 Metrics to Measure Performance

Accuracy and predictability were clearly defined by Gajos et al. [6] as metrics by which to evaluate adaptive interface performance. They define predictability is a measure of how easily the user can build a mental model of how the system will behave, and accuracy as a measure of the system's ability to correctly guess what the user needs [4]. A purely predictable interface would be static, while a high-accuracy interface would have UI elements placed in the adaptive space automatically when they are needed. There are clear trade-offs; the former offers no adaptive benefits, while the latter changes frequently and thus may seem unstable. Having considered this in prior work, Gajos et al. [6] hypothesized that greater predictability would increase user satisfaction and performance, while accuracy would have a smaller role. On the contrary, they found that accuracy had more influence than predictability on performance, utilization, and even, to a lesser extent, on satisfaction. This shows that adaptation can definitely be effective, but since the two metrics related to different aspects of the user experience, increasing one cannot make up for lowering the other [6].

4.3 User-As-Student Metaphor

Brusilovsky et al. [3] applied an incremental adaptation approach to InterBook, a web-based system for reading and writing adaptive electronic textbooks (AETs). Adaptive tutoring is accomplished through AETs. As the student reads, the system tracks the student's knowledge. It can then present new sections for reading on completion of any required prerequisites. Brusilovsky et al. treat the InterBook UI as a book to learn, with UI features corresponding to learnable concepts. In this way, they apply the same adaptive pattern to UI learning as AETs do to studying books, in what the authors call the user-as-student metaphor [3].

4.4 Special Needs Adaptation

Montague proposes the Shared User Model (SUM), a framework supporting custom adaptation [11]. SUM was embedded on an iPod touch with multiple environmental sensors. Combined with user logging, the authors evaluated benefits of SUM to disabled users. Garcia proposes the MyUI framework which supports modular development of adaptive software [8]. MyUI was designed to aid developers in building accessible adaptive user interfaces. At its core is the "adaptation engine" which determines how the system should behave given the user's needs. If the user has poor vision, the system will make onscreen text larger or read it aloud.

4.5 Split Menus and Visual Hints

Gajos et al. [7] explore multiple adaptive user interfaces of their own creation, based primarily on findings from prior work. In their 2006 work [7] they propose and evaluate three interfaces: a split interface, a moving interface, and a visual popout interface. All are compared to a baseline that is non-adaptive. The visual popout interface is similar to the baseline in that frequently used items do not move, but rather are highlighted in a bright color. If such an item is hidden inside a menu, the menu is highlighted as well. For evaluation, two kinds of tasks were performed: engaging, real-world activities; and less cognitively complex, albeit unrealistic, repetitive tasks. As adaptive interfaces have trade-offs, Gajos et al. considered costs and benefits of each interface. The moving interface results were above the baseline in some measures (e.g., physical demand), but below in others (e.g., control). The split interface achieved results consistently above both the baseline and the moving interface. Spatial stability was assumed to have been the greatest factor in the positive results for this high-benefit, low-cost option [7].

5. CHALLENGES

Adaptive interfaces are never perfect, but rather are evaluated in terms of trade-offs. A system that readily and accurately adapts can, in theory, result in enormous efficiency gains in the user's workflow. In practice, however, the user may get irritated and disoriented [1, 7]. Split menus are generally well received with good workflow improvements but UI elements, being copied rather than moved, take up extra screen real estate. Non-adaptive interfaces are most predictable, and the users appreciate the stability [4], but no benefits are gained. In addition, the development overhead required to build an adaptive interface can outweigh the perceived benefits. Garcia et al. addressed this, at least in part, with their MyUI framework [8].

6. RESEARCH FINDINGS AND FUTURE

With Gajos et al.'s split interface [7], by being able to use an item either in its original location or from the adaptive space, users felt they had a choice between two interfaces and this was appreciated. A button to toggle the adaptive approach on or off might also have the same effect, and would be interesting future work. Stability is found to be very important to users so they do not get lost and frustrated [1, 4, 7]. Further evaluation is needed in realistic scenarios. Existing systems seem promising, but evaluations such as Gajos et al.'s repetitive task [7] need to be made more realistic for useful results. Long term studies are difficult but necessary for gradual adaptation studies. As machine learning is currently a very active area of research [10], it should continually be investigated to improve accuracy of adaptive systems. Finally, developers and users alike need to be convinced that adaptive GUIs are worth the extra effort, a challenging but reachable goal.

7. REFERENCES

[1] Andreyev, S. (2012). Fixed Interfaces, Adaptive Interfaces... What is next? Total movability - a new paradigm for the user interface. *CoRR*, abs/1208.0408.

An overview of computing history with an emphasis on real-world interface examples and how they could be made adaptive. Several of the author's own opinions presented.

[2] Billsus, D., Brunk, C. A., Evans, C., Gladish, B. & Pazzani, M. (2002). Adaptive interfaces for ubiquitous web access. *Commun. ACM*, 45, 34-38. doi: 10.1145/506218.506240

Motivation for adaptive interfaces, especially on small screens, and the importance of implicitly learning about the user's needs.

[3] Brusilovsky, P., & Schwarz, E. (1997). User as student: Towards an adaptive interface for advanced web-based applications. *COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES*, 177-188.

An adaptive tutoring approach to teaching users how to use the online InterBook textbook tool.

[4] Findlater, L. & Gajos, K. Z. (2009). Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces. *AI Magazine*, 30, 68-73.

Exploration of the design space for adaptive GUIs and guidelines for their evaluation. Characteristics of adaptive algorithms and approaches that have been successful.

[5] Gajos, K. Z. (2008). *Automatically generating personalized user interfaces*. ProQuest.

Motivation for adaptive user interfaces with supporting experimental results in favor of their use.

[6] Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M. & Weld, D. S. (2008). Predictability and accuracy in adaptive user interfaces. *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (p./pp. 1271--1274), : ACM. ISBN: 9781605580111

A study applying the proposed predictability and accuracy metrics in evaluation of adaptive GUIs.

[7] Gajos, K. Z., Czerwinski, M., Tan, D. S. & Weld, D. S. (2006). Exploring the design space for adaptive graphical user interfaces. *AVI '06: Proceedings of the working conference on Advanced visual interfaces* (p./pp. 201-208), : ACM Press. ISBN: 1595933530

The evaluation of three different adaptive GUI paradigms with both real-world complex tasks and unrealistic repetitive tasks.

[8] García, A., Sánchez, J., Sánchez, V. & Hernández, J. A. (2012). Integration of a Regular Application into a User Interface Adaptation Engine in the MyUI Project.. In K. Miesenberger, A. I. Karshmer, P. Penaz & W. L. Zagler (eds.), ICCHP (1) (p./pp. 311-314), : Springer. ISBN: 978-3-642-31521-3

The MyUI project, a software framework allowing for easy development of adaptive user interfaces, and its “adaptation engine” which determines the most suitable interface for the user. The authors’ goal is ease-of-development of adaptive systems to support accessibility.

[9] Kardan, S. (2012). Data Mining for Adding Adaptive Interventions to Exploratory and Open-Ended Environments. In J. Masthoff, B. Mobasher, M. C. Desmarais & R. Nkambou (eds.), UMAP (p./pp. 365-368), : Springer. ISBN: 978-3-642-31453-7

A data mining approach to adding adaptive functionality to an exploratory environment.

[10] Langley, P. (1997). Machine Learning for Adaptive User Interfaces.. In G. Brewka, C. Habel & B. Nebel (eds.), KI (p./pp. 53-62), : Springer. ISBN: 3-540-63493-2

An overview of how machine learning approaches can be applied to adaptive personalization.

[11] Montague, K. (2012, October). Interactions speak louder than words: shared user models and adaptive interfaces. In *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology* (pp. 39-42). ACM.

Combining Shared User Models and adaptive approaches to create a more accessible interface for users with vision and motor impairments.